

z/VM  
7.3

*CP Commands and Utilities Reference*



**Note:**

Before you use this information and the product it supports, read the information in [“Notices” on page 2283.](#)

This edition applies to version 7, release 3 of IBM® z/VM® (product number 5741-A09) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2023-09-12

© **Copyright International Business Machines Corporation 1991, 2023.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Figures.....</b>	<b>xix</b>
<b>Tables.....</b>	<b>xxi</b>
<b>About This Document.....</b>	<b>XXV</b>
Intended Audience.....	xxv
Where to Find More Information.....	xxv
Links to Other Documents and Websites.....	xxv
<b>How to provide feedback to IBM.....</b>	<b>xxvii</b>
<b>Summary of Changes for z/VM: CP Commands and Utilities Reference.....</b>	<b>xxix</b>
SC24-6268-73, z/VM 7.3 (September 2023).....	xxix
SC24-6268-73, z/VM 7.3 (May 2023).....	xxix
SC24-6268-73, z/VM 7.3 (September 2022).....	xxx
SC24-6268-11, z/VM 7.2 (May 2022).....	xxxiii
SC24-6268-10, z/VM 7.2 (December 2021).....	xxxiv
SC24-6268-09, z/VM 7.2 (August 2021).....	xxxiv
SC24-6268-09, z/VM 7.2 (July 2021).....	xxxv
SC24-6268-08, z/VM 7.2 (March 2021).....	xxxvi
SC24-6268-07, z/VM 7.2 (December 2020).....	xxxviii
SC24-6268-06, z/VM 7.2 (September 2020).....	xxxix
<b>Chapter 1. Introduction.....</b>	<b>1</b>
Using Commands.....	1
Command Structure.....	2
The Command Name.....	2
The Command Operands.....	2
How to Enter Comments in the CP Command Language.....	2
Character Sets.....	2
Using Truncations and Abbreviations.....	3
Syntax, Message, and Response Conventions.....	4
Privilege Classes.....	6
Delayed Responses to CP Commands.....	7
Example.....	9
Using the Online HELP Facility.....	10
<b>Chapter 2. CP Commands.....</b>	<b>11</b>
ASTERISK (*).....	12
#CP.....	13
ACNT.....	15
ACTIVATE ISLINK.....	16
ADJUNCT.....	18
ASSOCIATE EXIT.....	20
ASSOCIATE MESSAGES / MSGS.....	24
AT.....	27
ATTACH.....	30
ATTN.....	49
AUTOLOG.....	50

BACKSPACE/BACKWARD.....	54
BEGIN.....	56
CHANGE.....	58
CLOSE.....	68
COMMANDS.....	78
COMMIT.....	80
CONCOPY TERMINATE.....	82
COUPLE.....	83
CP.....	88
CPACCESS.....	89
CPCACHE.....	94
CPHX.....	96
CPLISTFILE.....	99
CPRELEASE.....	103
CPTYPE.....	105
CPU.....	107
CPVLOAD.....	109
CPXLOAD.....	111
CPXUNLOAD.....	117
DEACTIVE CONV.....	119
DEACTIVE ISLINK.....	120
DEFINE.....	122
DEFINE ALIAS.....	127
DEFINE CFLINK.....	130
DEFINE CHPID / PATH.....	132
DEFINE COMMAND / CMD.....	143
DEFINE CONSOLE.....	149
DEFINE COWNED.....	150
DEFINE CPU.....	152
DEFINE CPUPOOL.....	156
DEFINE CRYPTO.....	157
DEFINE CTCA.....	159
DEFINE CU / CNTLUNIT.....	161
DEFINE DEVICE / IODEVICE.....	167
DEFINE DIAGNOSE.....	172
DEFINE EXIT.....	176
DEFINE GRAF.....	179
DEFINE HYPERPAVALIAS.....	180
DEFINE LAN.....	181
DEFINE LINE.....	186
DEFINE LPAR.....	187
DEFINE MDISK.....	188
DEFINE MSGPROC.....	191
DEFINE NIC.....	194
DEFINE PAVALIAS.....	198
DEFINE PCIFUNCTION.....	199
DEFINE RELODOMAIN.....	203
DEFINE RESPOOL.....	205
DEFINE (Spooling Device).....	209
DEFINE STORAGE.....	213
DEFINE SUBSTITUTE.....	222
DEFINE (Temporary Disk).....	223
DEFINE TIMEZONE.....	225
DEFINE (Virtual Device).....	227
DEFINE (Virtual Disk in Storage).....	230
DEFINE VSWITCH.....	232
DEFSEG.....	246
DEFSYS.....	253

DELETE CHPID / PATH.....	259
DELETE CPUPOOL.....	262
DELETE CU / CNTLUNIT.....	263
DELETE DEVICE / IODEVICE.....	265
DELETE EDEVICE.....	268
DELETE LPAR.....	270
DELETE PCIFUNCTION.....	272
DELETE RDEVICE.....	274
DELETE RESPOOL.....	276
DESTAGE.....	277
DETACH.....	279
DETACH CFLINK.....	282
DETACH CPU.....	284
DETACH CRYPTO.....	286
DETACH LAN.....	290
DETACH MSGPROC.....	292
DETACH NIC.....	293
DETACH PCIFUNCTION.....	294
DETACH (Real Device).....	296
DETACH SYSASCII.....	302
DETACH (Virtual Device).....	304
DETACH VSWITCH.....	307
DIAL.....	308
DISABLE (Device).....	311
DISABLE COMMAND / CMD.....	313
DISABLE DIAGNOSE.....	316
DISABLE EXITS.....	319
DISABLE HCD.....	321
DISASSOCIATE.....	324
DISCARD PINNED.....	327
DISCONNECT.....	329
DISPLAY.....	331
DISPLAY ACCESS LIST.....	335
DISPLAY (Guest Storage - ESA/XC, z/XC).....	337
DISPLAY (Guest Storage - ESA/390).....	345
DISPLAY (Guest Storage - z/Architecture).....	354
DISPLAY (Host Storage).....	363
DISPLAY LKS.....	368
DISPLAY PREFIX.....	372
DISPLAY PSW.....	373
DISPLAY PSWG.....	376
DISPLAY (Registers).....	378
DISPLAY SCHIB.....	383
DRAIN (Disk).....	386
DRAIN UR (Unit Record Device).....	388
DUMP.....	395
DUMP ACCESS LIST.....	399
DUMP (Guest Storage - ESA/XC, z/XC).....	401
DUMP (Guest Storage - ESA/390).....	408
DUMP (Guest Storage - z/Architecture).....	416
DUMP (Host Storage).....	424
DUMP LKS.....	429
DUMP PREFIX.....	433
DUMP PSW.....	434
DUMP PSWG.....	437
DUMP (Registers).....	439
DUMP SCHIB.....	445
DUPLEX.....	448

ECHO.....	457
ENABLE (Device).....	458
ENABLE COMMAND / CMD.....	460
ENABLE DIAGNOSE.....	463
ENABLE EXITS.....	465
EXPLORE FCP.....	467
EXTERNAL.....	469
FLASHCOPY.....	471
FLASHCOPY BACKGNDCOPY.....	480
FLASHCOPY ESTABLISH.....	483
FLASHCOPY RESYNC.....	489
FLASHCOPY TGTWRITE.....	492
FLASHCOPY WITHDRAW.....	494
FLUSH.....	498
FOR.....	501
FORCE.....	504
FORWARD.....	508
FREE LOGON.....	510
FREE PRINTER/PUNCH.....	511
GIVE.....	513
GIVE (Real Device).....	515
GIVE (Virtual Device).....	518
HALT.....	520
HOLD LOGON.....	522
HOLD PRINTER/PUNCH.....	523
HYPERSWAP.....	525
INDICATE.....	531
INDICATE ACTIVE.....	532
INDICATE I/O.....	534
INDICATE LOAD.....	535
INDICATE MULTITHREAD.....	539
INDICATE NSS.....	543
INDICATE PAGING.....	546
INDICATE QUEUES.....	548
INDICATE SPACES.....	552
INDICATE USER.....	555
IPL.....	566
LINK.....	578
LOADBUF.....	592
LOADVFCB.....	598
LOCATE.....	602
LOCATE CMDBK.....	603
LOCATE DGNBK.....	608
LOCATE FILID.....	610
LOCATE FRAMETBL.....	612
LOCATE ICLBK.....	614
LOCATE LDEV.....	616
LOCATE RDEV.....	618
LOCATE RPCI.....	621
LOCATE SHPBK.....	623
LOCATE SNABK.....	625
LOCATE SPFBK.....	627
LOCATE (Storage).....	630
LOCATE SXSTE.....	634
LOCATE SYMBOL.....	636
LOCATE VDEV.....	639
LOCATE VMDBK.....	642
LOCATE VPCI.....	645

LOCATE VSMBK.....	647
LOCATE XITBK.....	649
LOCATEVM.....	651
LOCK.....	654
LOGOFF.....	659
LOGON.....	662
MESSAGE.....	673
MESSAGE ALL.....	674
MESSAGE (User).....	676
MODIFY CHPID / PATH.....	679
MODIFY COMMAND / CMD.....	683
MODIFY CU / CNTLUNIT.....	688
MODIFY DEVICE / IODEVICE.....	693
MODIFY DIAGNOSE.....	697
MODIFY EXIT.....	700
MODIFY PCIFUNCTION.....	703
MODIFY PRIV_CLASSES.....	705
MONITOR.....	707
MONITOR EVENT.....	710
MONITOR SAMPLE.....	718
MONITOR START/STOP.....	727
MSGNOH.....	731
NOTREADY.....	734
ORDER.....	735
PER.....	739
PURGE.....	740
PURGE IMG.....	743
PURGE NLS.....	745
PURGE NSS.....	747
PURGE TRFILES.....	749
PURGE UCR.....	751
QUERY ABEND.....	753
QUERY ACCOUNT.....	755
QUERY ADJUNCT.....	756
QUERY AGELIST.....	757
QUERY ALL.....	759
QUERY ALLOC.....	773
QUERY BYUSER.....	781
QUERY CACHE.....	782
QUERY CACHEFW.....	784
QUERY CAPABILITY.....	786
QUERY CFLINKS.....	788
QUERY CHPID.....	791
QUERY CHPIDS.....	796
QUERY CHPIDV.....	798
QUERY CMDLIMIT.....	799
QUERY COLLECT.....	800
QUERY COMMANDS.....	802
QUERY CONCOPY.....	804
QUERY CONFIGMODE.....	806
QUERY CONTROLLER.....	808
QUERY CONV.....	810
QUERY CPCHECKING.....	816
QUERY CPCMDS.....	817
QUERY CPDISKS.....	824
QUERY CPLANGLIST.....	826
QUERY CPLANGUAGE.....	828
QUERY CPLEVEL.....	829

QUERY CLOAD.....	832
QUERY CPOWNER.....	834
QUERY CPPROTECT.....	836
QUERY CPSERVICE.....	839
QUERY CPTRACE.....	842
QUERY CPUAFFINITY.....	845
QUERY CPUID.....	846
QUERY CPUPOOL.....	847
QUERY CPXLOAD.....	848
QUERY CRYPTO.....	853
QUERY CTCA.....	857
QUERY CU.....	860
QUERY D8ONECMD.....	865
QUERY DASD.....	867
QUERY DASDFW.....	879
QUERY DATEFORMAT.....	881
QUERY DEVICES.....	883
QUERY DIAGNOSE.....	885
QUERY DUMP.....	887
QUERY DUMPDEV.....	890
QUERY DUPLEX.....	893
QUERY DYNAMIC_I/O.....	895
QUERY EDEVICE.....	898
QUERY ENCRYPT.....	904
QUERY EQID.....	905
QUERY EXITS.....	907
QUERY FCP.....	911
QUERY FENCES.....	915
QUERY FILES.....	917
QUERY FLASHCOPY.....	920
QUERY FRAMES.....	925
QUERY GATEWAY.....	929
QUERY GRAF.....	931
QUERY HCD.....	934
QUERY HOLD.....	936
QUERY HOTIO.....	938
QUERY HYPERSWAP.....	941
QUERY ICLNAME.....	943
QUERY IMG.....	945
QUERY IO_OPT.....	948
QUERY IOASSIST.....	950
QUERY IOPRIORITY.....	952
QUERY IPLPARMS.....	954
QUERY ISFC.....	956
QUERY ISLINK.....	959
QUERY IUCV.....	961
QUERY JOURNAL.....	963
QUERY KEYALIAS.....	964
QUERY LAN.....	966
QUERY LDEVS.....	973
QUERY LINES.....	976
QUERY LINKS.....	979
QUERY LKFAC.....	981
QUERY LKFACR.....	983
QUERY LOADDEV.....	985
QUERY LOGMSG.....	988
QUERY LPARS.....	991
QUERY LSYSTEM.....	995



QUERY MAXLDEV.....	996
QUERY MAXSPOOL.....	997
QUERY MAXUSERS.....	999
QUERY MDCACHE.....	1000
QUERY MDISK.....	1004
QUERY MEMASSIST.....	1008
QUERY MITIME.....	1011
QUERY MONDATA.....	1013
QUERY MONITOR.....	1014
QUERY MSS.....	1025
QUERY MULTITHREAD.....	1026
QUERY NAMES.....	1029
QUERY NEW_DEVICES.....	1033
QUERY NIC.....	1034
QUERY NLS.....	1038
QUERY NSS.....	1041
QUERY NVS.....	1050
QUERY OBSERVER.....	1052
QUERY OSA.....	1055
QUERY PAGING.....	1060
QUERY PASSWORD.....	1062
QUERY PATHS.....	1064
QUERY PAV.....	1068
QUERY PCIFUNCTION.....	1070
QUERY PENDING COMMANDS.....	1074
QUERY PER.....	1076
QUERY PFnn.....	1077
QUERY PINNED.....	1079
QUERY PORT.....	1082
QUERY (Printer ID).....	1089
QUERY PRIVCLASS.....	1090
QUERY PROCESSORS.....	1092
QUERY PRODUCT.....	1095
QUERY PROMPT.....	1097
QUERY PSWTRANS.....	1099
QUERY QIOASSIST.....	1100
QUERY QUICKDSP.....	1102
QUERY READER/PRINTER/PUNCH.....	1103
QUERY (Real Device).....	1114
QUERY RECORDING.....	1130
QUERY RELODOMAIN.....	1132
QUERY REORDER.....	1134
QUERY RESERVED.....	1135
QUERY RESOURCE.....	1138
QUERY RESPOOL.....	1140
QUERY RETRIEVE.....	1143
QUERY RSAW.....	1145
QUERY SCMBKS.....	1147
QUERY SCMEASURE.....	1149
QUERY SCREEN.....	1151
QUERY SDF.....	1152
QUERY SECUSER.....	1153
QUERY SET.....	1156
QUERY SHARE.....	1161
QUERY SHUTDOWN.....	1163
QUERY SHUTDOWNTIME.....	1166
QUERY SIGNAL.....	1167
QUERY SIGNALS.....	1168

QUERY SPACES.....	1170
QUERY SRM.....	1173
QUERY SSI.....	1177
QUERY STGEXEMPT.....	1183
QUERY STGLIMIT.....	1185
QUERY STORAGE/STORE.....	1186
QUERY STP.....	1191
QUERY SUBSTITUTE.....	1193
QUERY SWITCHES.....	1194
QUERY SXSPAGES.....	1196
QUERY SXSSTORAGE.....	1199
QUERY SYSASCII.....	1200
QUERY SYSCONTROL.....	1201
QUERY SYSOPER.....	1202
QUERY SYSTEM.....	1203
QUERY TAG.....	1205
QUERY TAPES.....	1207
QUERY TDISK.....	1211
QUERY TDISKCLR.....	1212
QUERY TERMINAL.....	1213
QUERY THROTTLE.....	1217
QUERY TIME.....	1219
QUERY TIMEZONES.....	1222
QUERY TOKEN.....	1224
QUERY TRACE.....	1226
QUERY TRACEFRAMES.....	1230
QUERY TRFILES.....	1231
QUERY TRSAVE.....	1235
QUERY TRSOURCE.....	1239
QUERY UCR.....	1246
QUERY UNDERSCORE.....	1249
QUERY UNRESOLVED.....	1250
QUERY UR.....	1252
QUERY USERID.....	1261
QUERY USERS.....	1264
QUERY VARIABLE.....	1267
QUERY VCONFIG.....	1270
QUERY VDISK.....	1271
QUERY VIRTUAL ALL.....	1273
QUERY VIRTUAL CHPID.....	1275
QUERY VIRTUAL CONSOLE.....	1278
QUERY VIRTUAL CPUS.....	1282
QUERY VIRTUAL CRYPTO.....	1284
QUERY VIRTUAL CTCA.....	1287
QUERY VIRTUAL DASD.....	1288
QUERY (Virtual Device).....	1291
QUERY VIRTUAL DUPLEX.....	1295
QUERY VIRTUAL FCP.....	1297
QUERY VIRTUAL FLASHCOPY.....	1300
QUERY VIRTUAL GRAF.....	1303
QUERY VIRTUAL LINES.....	1305
QUERY VIRTUAL MSGDEVICES.....	1307
QUERY VIRTUAL MSGPROC.....	1308
QUERY VIRTUAL NIC.....	1310
QUERY VIRTUAL OSA.....	1316
QUERY VIRTUAL PAV.....	1319
QUERY VIRTUAL PCIFUNCTION.....	1322
QUERY VIRTUAL PRINTER.....	1324

QUERY VIRTUAL PUNCH.....	1328
QUERY VIRTUAL READER.....	1331
QUERY VIRTUAL STORAGE/STORE.....	1333
QUERY VIRTUAL SWITCHES.....	1336
QUERY VIRTUAL SYSASCII.....	1337
QUERY VIRTUAL TAPES.....	1338
QUERY VIRTUAL UR.....	1340
QUERY VMDUMP.....	1341
QUERY VMLAN.....	1344
QUERY VMRELOCATE.....	1348
QUERY VMSG/PVMSG.....	1349
QUERY VSWITCH.....	1351
QUERY VTOD.....	1370
QUERY WRKALLEG.....	1372
READY.....	1374
RECORDING.....	1375
REDEFINE.....	1378
REFRESH.....	1380
RELSPACE.....	1382
REPEAT.....	1384
REQUEST.....	1386
RESET.....	1387
RESTART MSGPROC.....	1391
REWIND.....	1393
SAVESEG.....	1394
SAVESYS.....	1397
SCHEDULE.....	1401
SCREEN.....	1404
SEND.....	1407
SET.....	1411
SET ABEND.....	1413
SET ACCOUNT.....	1416
SET ADJUNCTS.....	1417
SET AGELIST.....	1418
SET AUTOPOLL.....	1421
SET CACHE.....	1422
SET CACHEFW.....	1427
SET CFLINK.....	1430
SET CMDLIMIT.....	1432
SET CONCEAL.....	1434
SET CONFIGMODE.....	1436
SET CPCHECKING.....	1439
SET CPCONIO.....	1440
SET CPLANGUAGE.....	1441
SET CPPROTECT.....	1443
SET CPTRACE.....	1446
SET CPUAFFINITY.....	1472
SET CPUID.....	1474
SET CPUPOOL.....	1475
SET CRYPTO.....	1476
SET CU.....	1478
SET D8ONECMD.....	1482
SET DASD.....	1484
SET DASDFW.....	1485
SET DATEFORMAT.....	1488
SET DEVICES.....	1490
SET DIALDROP.....	1492
SET DUMP.....	1493

SET DUMPDEV.....	1496
SET DYNAMIC_I/O.....	1502
SET EDEVICE.....	1504
SET EMSG.....	1512
SET ENCRYPT.....	1513
SET HOTIO.....	1516
SET IMSG.....	1519
SET IO_OPT.....	1521
SET IOASSIST.....	1523
SET IOCDS_ACTIVE.....	1525
SET IOPRIORITY.....	1527
SET IPLPARMS.....	1530
SET JOURNAL.....	1533
SET KEYALIAS.....	1534
SET LAN.....	1536
SET LINEDIT.....	1539
SET LKFAC.....	1540
SET LKFACR.....	1542
SET LOADDEV.....	1544
SET LOGMSG.....	1551
SET LSYSTEM.....	1553
SET MACHINE.....	1555
SET MAXLDEV.....	1558
SET MAXUSERS.....	1559
SET MDCACHE.....	1561
SET MEMASSIST.....	1568
SET MIH.....	1570
SET MITIME.....	1571
SET MODE.....	1574
SET MONDATA.....	1575
SET MSG.....	1576
SET MSGFACIL.....	1577
SET MULTITHREAD.....	1578
SET NEW_DEVICES.....	1581
SET NIC.....	1583
SET NOPDATA.....	1587
SET NVS.....	1588
SET OBSERVER.....	1591
SET PAGEX.....	1594
SET PAGING.....	1596
SET PASSWORD.....	1598
SET PCIFUNCTION.....	1600
SET PFnn.....	1602
SET PFnn COPY.....	1605
SET PFnn RETRIEVE.....	1607
SET PFnn TAB.....	1609
SET PORT GROUP.....	1610
SET PRIVCLASS.....	1616
SET PRODUCT.....	1619
SET PROMPT.....	1621
SET PSWTRANS.....	1622
SET QIOASSIST.....	1623
SET QUICKDSP.....	1625
SET RDEVICE.....	1627
SET RDEVICE Advanced Function Printers.....	1629
SET RDEVICE Card Punches.....	1631
SET RDEVICE Card Readers.....	1634
SET RDEVICE CLEAR.....	1636

SET RDEVICE Communication Controllers.....	1638
SET RDEVICE DASD.....	1641
SET RDEVICE Graphic Display Devices.....	1644
SET RDEVICE Impact Printers.....	1648
SET RDEVICE Integrated Communication Adapters.....	1653
SET RDEVICE Special Devices.....	1655
SET RDEVICE Tape Units.....	1658
SET RDEVICE Terminals.....	1661
SET RDEVICE Unsupported Devices.....	1663
SET RDEVICE 3800 Printers.....	1666
SET RECORD.....	1671
SET REORDER.....	1672
SET RESERVED.....	1674
SET RESPOOL.....	1679
SET RETRIEVE.....	1681
SET RUN.....	1683
SET SCMEASURE.....	1685
SET SECUSER.....	1688
SET SHARE.....	1691
SET SHARED.....	1696
SET SHUTDOWNTIME.....	1698
SET SHUTSIGNAL.....	1700
SET SIGNAL.....	1701
SET SMSG.....	1703
SET SRM.....	1704
SET SSI.....	1714
SET STGEXEMPT.....	1719
SET STGLIMIT.....	1721
SET STORAGE.....	1722
SET SVC76.....	1729
SET SYSCONTROL.....	1730
SET SYSOPER.....	1731
SET TAPE.....	1733
SET THROTTLE.....	1734
SET TIMEBOMB.....	1736
SET TIMER.....	1737
SET TIMEZONE.....	1738
SET TOKEN.....	1740
SET TRACEFRAMES.....	1742
SET UNDERSCORE.....	1744
SET VARIABLE.....	1745
SET VCONFIG.....	1747
SET VDISK.....	1749
SET VMCONIO.....	1751
SET VMLAN.....	1752
SET VMRELOCATE.....	1755
SET VSWITCH.....	1757
SET VTOD.....	1781
SET WNG.....	1784
SET WRKALLEG.....	1785
SET 370ACCOM.....	1787
SHUTDOWN.....	1789
SIGNAL.....	1796
SILENTLY.....	1798
SLEEP.....	1799
SMSG.....	1800
SNAPDUMP.....	1802
SPACE.....	1804

SPOOL.....	1806
SPXTAPE.....	1818
SPXTAPE CANCEL.....	1834
SPXTAPE DUMP.....	1836
SPXTAPE END.....	1844
SPXTAPE LOAD.....	1846
SPXTAPE SCAN.....	1857
START (Disk).....	1864
START UR (Unit Record).....	1866
STOP.....	1878
STORE.....	1879
STORE (Guest Storage - ESA/XC, z/XC).....	1881
STORE (Guest Storage - ESA/390).....	1886
STORE (Guest Storage - z/Architecture).....	1892
STORE (Host Storage).....	1898
STORE PSW.....	1901
STORE PSWA.....	1903
STORE PSWG.....	1904
STORE (Registers).....	1905
STORE STATUS.....	1908
SYNCDMS.....	1910
SYSTEM.....	1911
TAG.....	1914
TERMINAL.....	1917
TERMINAL APL.....	1919
TERMINAL ASCIITBL.....	1920
TERMINAL ATTN.....	1921
TERMINAL AUTOOCR.....	1922
TERMINAL BREAKIN.....	1923
TERMINAL BRKKEY.....	1924
TERMINAL CHARDEL.....	1926
TERMINAL CNTL.....	1927
TERMINAL CONMODE.....	1928
TERMINAL ESCAPE.....	1930
TERMINAL HILIGHT.....	1931
TERMINAL HOLD.....	1932
TERMINAL LINEDEL.....	1933
TERMINAL LINEND.....	1934
TERMINAL LINESIZE.....	1935
TERMINAL MODE.....	1936
TERMINAL MORE.....	1937
TERMINAL PRECEDENCE.....	1938
TERMINAL PROMPT.....	1939
TERMINAL SCROLL.....	1940
TERMINAL SYS3270.....	1941
TERMINAL TABCHAR.....	1942
TERMINAL TEXT.....	1943
TERMINAL TIMESTAMP.....	1944
TERMINAL TYPE.....	1945
TRACE.....	1946
TRACE: Options.....	1949
TRACE: Command Responses.....	1954
TRACE AIF.....	1963
TRACE ALL.....	1964
TRACE APPEND.....	1965
TRACE AR.....	1966
TRACE BRANCH.....	1968
TRACE CLEAR.....	1971

TRACE COUNT.....	1972
TRACE DELETE.....	1973
TRACE END.....	1974
TRACE EXTERNAL/PROGRAM.....	1975
TRACE G.....	1976
TRACE GG.....	1978
TRACE GOTO/CALL/RETURN.....	1980
TRACE INSTRUCTION.....	1982
TRACE I/O.....	1983
TRACE MCH.....	1985
TRACE mnemonic1.....	1986
TRACE mnemonic2.....	1988
TRACE RIO/RSCH/SIO/SIOF/SSCH.....	1989
TRACE STORE (ESA/390, z/Architecture).....	1991
TRACE STORE (ESA/XC, z/XC).....	1995
TRACE SVC/DIAGNOSE/MC.....	1998
TRACE TABLE.....	1999
TRACE TPI.....	2001
TRACE TRAP.....	2002
TRACE TXSUSPEND/NOTXSUSPEND.....	2003
TRANSFER.....	2005
TRSAVE.....	2011
TRSOURCE.....	2018
TRSOURCE DISABLE.....	2022
TRSOURCE DISPLAY.....	2023
TRSOURCE DROP.....	2024
TRSOURCE ENABLE.....	2025
TRSOURCE ID (Definition Operands).....	2026
UNCOUPLE.....	2040
UNDIAL.....	2042
UNLOCK.....	2043
VARY CHPID.....	2046
VARY CORE.....	2048
VARY CRYPTO.....	2050
VARY PATH.....	2053
VARY PCIFUNCTION.....	2059
VARY PROCESSOR.....	2061
VARY (Real Device).....	2063
VARY SUBCHANNEL.....	2067
VDELETE.....	2069
VINPUT.....	2071
VMDUMP.....	2072
VMRELOCATE.....	2083
WARNING.....	2097
XAUTOLOG.....	2100
XLINK.....	2108
XLINK CHECK.....	2110
XLINK FINDTAB.....	2111
XLINK RESET.....	2112
<b>Chapter 3. CP Utilities.....</b>	<b>2113</b>
CCDUMP.....	2114
CCLOAD.....	2116
CPEREPXA.....	2118
CPFMTXA.....	2124
CPSYNTAX.....	2143
CVTOVRID.....	2147

DDR.....	2149
DIRECTXA.....	2168
DISKMAP.....	2178
DUMpload.....	2180
DUMPLD2.....	2184
FORMSSI.....	2189
GENIMAGE.....	2192
HCPLDR.....	2195
IMAGELIB.....	2201
IOCP.....	2203
IOEXPLOR.....	2205
IPWIZARD.....	2210
MIGR51D.....	2211
MONWRITE.....	2212
MONWSTOP.....	2216
MOVE2SFS.....	2217
RETRIEVE.....	2219
SALIPL.....	2222
SCSIDISC.....	2228
SDINST.....	2235
SECTABLE.....	2239
SFCOUNTR.....	2241
SPORDER.....	2243
TRACERED.....	2244
UTILITY.....	2255
XLINK DISPLAY.....	2257
XLINK FORMAT.....	2260

**Appendix A. Restricted User IDs..... 2263**

**Appendix B. Logical Line Editing Symbols..... 2267**

Logical Character Delete.....	2267
Logical Line End.....	2267
Logical Line Delete.....	2267
Logical Escape.....	2268
Defining Logical Line Editing Symbols.....	2268

**Appendix C. ASCII Translation..... 2269**

Provision of Translate Tables.....	2269
ASCII to EBCDIC Differences.....	2269
EBCDIC to ASCII Differences.....	2269

**Appendix D. HCPLDR Control Statements and Output..... 2271**

Loader Control Statements.....	2271
Printer Control Statement.....	2271
Set Page Boundary Control Statement.....	2271
Unconditional Page Boundary Control Statement.....	2272
Conditional Page Boundary Control Statement.....	2272
Set Location Counter Control Statement.....	2272
Padding Control Statement.....	2273
Parameter Control Statement.....	2273
Subsystem Control Statement.....	2274
Loader Termination Control Statement.....	2274
Replace Control Statement.....	2274
Verify Control Statement.....	2275
Include Control Section Control Statement.....	2276
Delete Control Statement.....	2276



Output.....	2277
Text Deck Information.....	2277
System Summary.....	2277
CSECTs with Size Greater than Conditional Page Boundary.....	2277
External Cross Reference: Symbol to Adcon CSECT.....	2278
Cross Reference: CSECT to Referenced Symbol.....	2278
CSECT Cross Reference: Referenced CSECT to Adcon Location.....	2278
CSECT Cross Reference: Adcon Location to Referenced CSECT.....	2279
Assembly Dates and Entry Points.....	2279
CSECT Assembly Dates, Least to Most Recent.....	2279
Load Map Sorted by Name.....	2280
Load Map Sorted by Address.....	2281
Other Information.....	2281
SMALLMAP File Information.....	2282
<b>Notices.....</b>	<b>2283</b>
Trademarks.....	2284
Terms and Conditions for Product Documentation.....	2284
IBM Online Privacy Statement.....	2285
<b>Bibliography.....</b>	<b>2287</b>
Where to Get z/VM Information.....	2287
z/VM Base Library.....	2287
z/VM Facilities and Features.....	2288
Prerequisite Products.....	2290
Related Products.....	2290
Other Related Publications.....	2290
<b>Index.....</b>	<b>2291</b>



---

# Figures

1. An Annotated Sample of Output from the TYPE and PRINT Functions of the DDR Program.....	2163
2. Formatted CP Trace Entry.....	2250
3. Guest Trace Data of Type GCS in Event Mode.....	2250
4. Formatted Guest Trace Data of Type TSAF in Block mode.....	2250
5. I/O Data Formatted by TRACERED.....	2251
6. Data Trace Entry Output Formatted by TRACERED.....	2251
7. CP Trace Table Output.....	2252
8. Hex Guest Trace Data of Type TSAF in Block Mode.....	2252
9. Unformatted Guest Trace Output.....	2252
10. System Summary for CP.....	2277
11. CSECTs with Size Greater than Conditional Page Boundary.....	2278
12. External Cross Reference.....	2278
13. CSECT to Referenced Symbol Report.....	2278
14. Referenced CSECT to Adcon.....	2279
15. Adcon Location to Referenced CSECT.....	2279
16. Assembly Dates and Entry Points.....	2279
17. Assembly Dates, Least to Most Recent.....	2280
18. Load Map Sorted by Name.....	2280
19. VERIFY Failures, Unreferenced CSECTs, and Relocation Summary.....	2281
20. Contents of SMALLMAP option.....	2282



---

# Tables

1. Character Set Summary.....	3
2. Examples of Syntax Diagram Conventions.....	4
3. Privilege Classes.....	7
4. System Responses to Various #CP Commands.....	13
5. DASD Control Levels.....	36
6. Example 1A - Crypto Configuration.....	39
7. Example 1B - Crypto Configuration.....	39
8. Example 1C - Crypto Configuration.....	40
9. Example 2A - Crypto Configuration.....	40
10. Example 2B - Crypto Configuration.....	41
11. Effect of File Status, Device Options, and CLOSE Command Options on Reader Files.....	74
12. Maximum Number of Logical Partition Names.....	140
13. ESA/390 or General Logical Partition.....	153
14. LINUX Only Logical Partition.....	153
15. z/VM Logical Partition.....	153
16. Maximum Number of Logical Partition Names.....	201
17. Maximum Input Values for Storage Units.....	215
18. Source-target pairs supported by FLASHCOPY.....	472
19. FlashCopy Resync Operation.....	489
20. Storage Units of Measure.....	553
21. Storage Units of Measure.....	557
22. LOADDEV and DUMPDEV parameters.....	571
23. LOADDEV and DUMPDEV parameters requirements by type of list-directed IPL.....	572

24. Linking to a Device in READ Mode.....	581
25. Linking to a Device in WRITE Mode.....	582
26. Linking to a Device in Multiple WRITE Mode.....	582
27. Link Conflict Analysis When Virtual Reserve/Release is Authorized.....	584
28. Link Conflict Analysis When Simulated Working Allegiance is Active.....	585
29. Page Number Derivation.....	654
30. Maximum Input Values for Storage Units.....	664
31. Valid Classes and Types for Seeks and I/O Domains.....	707
32. QUERY FLASHCOPY HARDWARE FLGS Field.....	922
33. Issuing the SEND Command with or without the CP Operand.....	1408
34. DUMPDEV parameters requirements by type of guest IPL with dump.....	1499
35. IPL Parameters.....	1530
36. LOADDEV parameters requirements by type of list-directed IPL without dump.....	1547
37. Virtual Machine Architecture Modes.....	1555
38. Maximum Input Values for Storage Units.....	1723
39. vidset Ranges.....	1761
40. portset Ranges.....	1763
41. Disposition of Reader Files Depending upon HOLD/KEEP Settings.....	1815
42. SPXTAPE LOAD Recovery Matrix.....	1854
43. Non-z/Architecture mode guest.....	1908
44. z/Architecture mode guest.....	1908
45. Time Stamp Prefix Sources.....	1944
46. TRACE Options Valid in Which Modes.....	1950
47. Machine Check Codes.....	1958
48. Default File Names and File Types for System Trace Files.....	2014

49. Maximum Input Values for Storage Units.....	2101
50. Record format for SECTABLE FILE.....	2239
51. Load Map Contents.....	2280





# About This Document

---

This reference document lists and describes the IBM z/VM 7.3 control program (CP) commands and utilities for users of every privilege class.

## Intended Audience

---

This document is written for any and all users of CP commands. It is intended to help CP command users operate more effectively.

The descriptions of the commands and utilities contain the following types of information:

- Command or utility name
- Format and syntax
- User privilege class
- Purpose statement
- Operand descriptions
- Usage notes
- Responses
- Messages

## Where to Find More Information

---

For information about related publications, see the [“Bibliography” on page 2287](#).

## Links to Other Documents and Websites

The PDF version of this document contains links to other documents and websites. A link from this document to another document works only when both documents are in the same directory or database, and a link to a website works only if you have access to the Internet. A document link is to a specific edition. If a new edition of a linked document has been published since the publication of this document, the linked document might not be the latest edition.



# How to provide feedback to IBM

---

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. See [How to send feedback to IBM](#) for additional information.



# Summary of Changes for z/VM: CP Commands and Utilities Reference

---

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line (|) to the left of the change.

This publication is updated and published with greater frequency than user assistance (online help) is updated and delivered. If user assistance contains information that does not match the information in this publication, the publication contains the latest information.

## SC24-6268-73, z/VM 7.3 (September 2023)

---

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.3.

### [VM66678, VM66709] Warning Track Interruption Facility

With the PTFs for APARs VM66678 (CP) and VM66709 (Performance Toolkit), z/VM 7.3 exploits a feature of Processor Resource/Systems Manager (PR/SM) called the *warning-track-interruption facility*. z/VM's exploitation of this facility helps improve guest response time and overall performance of workloads that are run on vertical-low or vertical-medium logical processors.

The following CP commands are updated:

- “[QUERY SRM](#)” on page 1173
- “[SET SRM](#)” on page 1704

### [VM66673] Information update about Large Guest Reset Time Mitigation

In support of APAR VM66673, the descriptions of the following commands are updated to explain how multiple instances of large-memory resets are shown in responses to each command:

- “[INDICATE PAGING](#)” on page 546
- “[LOCATE VMDBK](#)” on page 642
- “[QUERY NAMES](#)” on page 1029

### [VM66683] Channel Path Usage

In support of APAR VM66683, a clarification about channel path usage is added to the description of this CP command:

- “[DEFINE NIC](#)” on page 194

## SC24-6268-73, z/VM 7.3 (May 2023)

---

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.3.

### [VM66423] Crypto Stateless-Command Filtering

With the PTF for APAR VM66423, z/VM 7.3 provides support for a capability of the Crypto Express8S adapter, when configured in Common Cryptographic Architecture (CCA) coprocessor mode, to enforce restrictions on classes of requests. This support is intended to be used to limit operations on shared crypto resources, such as prohibiting secure-key functions. z/VM's exploitation of this hardware capability

will replace the crypto stateless command filtering performed by z/VM for APVIRT crypto users, thus eliminating the need for future z/VM updates to allow stateless CCA operations beyond those currently supported. The z/VM hypervisor specifies the policy, and the Crypto Express8S adapter enforces it. Crypto Stateless-Command Filtering requires an IBM z16 family server with driver D51C Bundle S19 applied.

The following CP message is updated:

- HCP1711I

### **[VM66423] Removal of obsolete IOCP Parameters DYN and NODYN**

With the PTF for APAR VM66423, the IOCP parameters DYN and NODYN, which previously allowed or prevented dynamic changes to I/O configurations, no longer have any effect.

The following topics are updated:

- [“IOCP” on page 2203](#)
- [“SET IOCD\\_ACTIVE” on page 1525](#)
- [“SET TOKEN” on page 1740](#)

### **[VM66424, VM66434, VM66650] Guest Secure IPL Support**

With the PTF for APARs VM66424, VM66434, and VM66650, z/VM V7.3 supports guest secure IPL (load and dump) for both ECKD and SCSI devices. A z/VM user can request that the machine loader validate the signed IPL code by using the security keys that were previously loaded by the customer into the HMC certificate store. The validation ensures that the IPL code is intact, unaltered, and originates from a trusted build-time source.

Support is provided for the following guest operating systems:

- This support provides the ability for a Linux guest to exploit hardware to validate the code being booted, helping to ensure it is signed by the client or its supplier.

Linux on IBM zSystems instances that previously were able to perform secure boot first level on an IBM z15 or IBM LinuxONE III prior to Driver D41C Bundle S73a, or an IBM z16 or IBM LinuxONE 4 prior to Driver D51C Bundle S18, will no longer be able to use secure boot until appropriate additional support is applied to the Linux image. See [230428 Machine Alert for 8561, 8562, 3931, 3932 \(https://www-40.ibm.com/servers/resourcelink/lib03020.nsf/pagesByDocid/272B3DD994A65B538525899F005FA0E6?OpenDocument\)](https://www-40.ibm.com/servers/resourcelink/lib03020.nsf/pagesByDocid/272B3DD994A65B538525899F005FA0E6?OpenDocument) for details about the required service level of Linux to properly IPL securely first or second level after driver D41C Bundle S73a or Driver D51C Bundle S18 has been applied.

- z/OS® is supported in audit mode only. Full exploitation requires Virtual Flash Memory support, which is not available to a guest. In audit mode, the IPL code is checked but the IPL continues even if the code is not valid.

z/VM and the z/VM stand-alone dump utility do not support performing host IPL via List-Directed IPL (LD-IPL) from ECKD. In addition, Secure IPL of the z/VM host and z/VM stand-alone dump are not supported.

The following topics are updated:

- [“DEFSYS” on page 253](#)
- [“INDICATE USER” on page 555](#)
- [“IPL” on page 566](#)
- [“LOGON” on page 662](#)
- [“QUERY DUMPDEV” on page 890](#)
- [“QUERY LOADDEV” on page 985](#)
- [“SET DUMPDEV” on page 1496](#)
- [“SET LOADDEV” on page 1544](#)

- [“VMRELOCATE” on page 2083](#)

### **[VM66654] Thin Stack Adapter Interrupt Support**

With the PTF for APAR VM66654, z/VM 7.3 enhances the Thin Stack FCP SCSI device driver and associated I/O abstraction layers to exploit adapter interruptions and to implement improved error handling and reporting.

The following utility is updated:

- [“SCSIDISC” on page 2228](#)

### **[VM66673] Large Guest Reset Time Mitigation**

With the PTF for APAR VM66673, z/VM 7.3 provides support for up to 2TB virtual memory within a single virtual machine under a set of restrictions while also providing a mitigation for long logoff times of guests with a large amount of instantiated virtual memory.

The following commands are updated:

- [“INDICATE PAGING” on page 546](#)
- [“LOCATE VMDBK” on page 642](#)
- [“LOGOFF” on page 659](#)
- [“QUERY NAMES” on page 1029](#)

### **[VM66675] Enable larger (>64GB) NVMe paging allocation extents**

With the PTF for APAR VM66675, z/VM 7.3 allows paging space to be allocated anywhere on an NVMe EDEVICE and to be of any size up to the available capacity. This PTF also adds the ability for the CPFMTXA utility to accept FBA allocations specified in factors of megabytes, gigabytes, and terabytes.

See [“CPFMTXA” on page 2124](#).

This support requires the corresponding ICKDSF PTF for APAR PH52513, which removes the 64G restriction for PAGE allocation on an FBA volume.

### **[VM66677] Increase crashkernel area size**

With the PTFs for APAR VM66677, z/VM 7.2 and 7.3 increase the size of crashkernel area allocated by stand-alone dump programs in preparation for a future enhancement.

The following CP utility is updated:

- [“SDINST” on page 2235](#)

### **Miscellaneous updates for May 2023**

The following commands are updated:

- [“QUERY MONITOR” on page 1014](#): New response for next sample date and time.
- [“LOCATE VMDBK” on page 642](#): New operand \*\* to locate all VMDBKs on the system.

## **SC24-6268-73, z/VM 7.3 (September 2022)**

---

This edition supports the general availability of z/VM 7.3. Note that the publication number suffix (-73) indicates the z/VM release to which this edition applies.

## Eight-member SSI support

This support increases the maximum size of a single system image (SSI) cluster from four members to eight, enabling clients to grow their SSI clusters to allow for increased workloads and providing more flexibility to use live guest relocation (LGR) for nondisruptive upgrades and workload balancing.

The following CP commands are updated:

- [“DEFINE RELODOMAIN” on page 203](#)
- [“QUERY SSI” on page 1177](#)
- [“SET SSI” on page 1714](#)

The following CP utility is new:

- [“SFCOUNTR” on page 2241](#)

The following CP utility is updated:

- [“DIRECTXA” on page 2168](#)

## NVMe emulated device (EDEVICE) support

NVMe devices that are connected through PCI Express (PCIe) adapters can be defined and managed as Fixed-Block Architecture (FBA) EDEVICES. As such, all host and guest FBA functions are supported except for those functions that require stand-alone support such as warm start and checkpoint. Linux guests that exploit EDEVICES that are defined on NVMe adapters are not eligible for live guest relocation. NVMe adapters are only available on LinuxONE servers. For more information, see [Defining and Managing NVMe Devices in z/VM: CP Planning and Administration](#).

Documentation is updated for the following commands:

- [“ATTACH” on page 30 \(Messages\)](#)
- [“DELETE EDEVICE” on page 268 \(Usage notes, messages\)](#)
- [“QUERY CU” on page 860 \(Purpose\)](#)
- [“QUERY EDEVICE” on page 898 \(Usage notes, responses\)](#)
- [“QUERY PCIFUNCTION” on page 1070 \(Responses\)](#)
- [“RESET” on page 1387 \(Usage note\)](#)
- [“SET CPTRACE” on page 1446 \(Codes\)](#)
- [“SET CU” on page 1478 \(Purpose, usage notes\)](#)
- [“SET DUMP” on page 1493 \(Restriction for NVMe devices\)](#)
- [“SET EDEVICE” on page 1504 \(New operands, purpose, messages\)](#)

The following command is new:

- [“SET PCIFUNCTION” on page 1600](#)

## Display mnemonics for instructions that are not supported by CP Trace

z/VM 7.3 displays mnemonics for instructions in CP TRACE output regardless of whether the instruction is fully supported by z/VM. The following topics are updated:

- [“TRACE” on page 1946](#)
- [“TRACE: Command Responses” on page 1954](#)
- [“TRSOURCE” on page 2018](#)

## Miscellaneous updates for z/VM 7.3

Documentation for the following CP commands is updated:

- [“CPXLOAD” on page 111](#)



- [“DEFINE MDISK” on page 188](#)
- [“IOEXPLOR” on page 2205](#)
- [“QUERY DUMPDEV” on page 890](#)
- [“QUERY LOADDEV” on page 985](#)
- [“QUERY VIRTUAL DASD” on page 1288](#)
- [“SET CRYPTO” on page 1476](#)
- [“SET VTOD” on page 1781](#)
- Support for 4-character time zones:
  - [“DEFINE TIMEZONE” on page 225](#)
  - [“QUERY TIME” on page 1219](#)
  - [“QUERY TIMEZONES” on page 1222](#)
  - [“SET TIMEZONE” on page 1738](#)

The following CP utilities are updated:

- [“MIGR51D” on page 2211](#)
- [“SDINST” on page 2235](#)

## SC24-6268-11, z/VM 7.2 (May 2022)

---

This edition includes changes to support product changes provided or announced after the general availability of z/VM 7.2.

### **[VM66532] z/VM Support for IBM z16**

With the PTF for APAR VM66532, z/VM 7.2 provides support to enable guests to exploit function on IBM z16. The following support is included:

- Breaking-event-address register (BEAR) enhancement facility, which facilitates the debug of wild branches.
- Reset DAT protection facility, which provides a more efficient way to disable DAT protection, such as during copy-on-write or page-change tracking operations.
- RoCE Express3 adapter, which allows guests to exploit Routable RoCE, Zero Touch RoCE, and SMC-R V2 support.
- The Crypto Express8S (CEX8S) adapter, supported as a dedicated or shared resource. Dedicated guests are able to take advantage of all functions available with the CEX8S adapters, including assorted new enhancements and use of Quantum-Safe APIs. All crypto adapters that are configured in EP11 mode are reported with the 'P' suffix instead of the 'S' suffix.

The following topics are updated:

- [“DUMP \(Registers\)” on page 439](#)
- [“QUERY CRYPTO” on page 853](#)
- [“QUERY VIRTUAL CRYPTO” on page 1284](#)
- [“STORE STATUS” on page 1908](#)
- [“TRACE mnemonic1” on page 1986](#)

### **[VM66534] Host Exploitation of Crypto Interruptions**

With the PTF for APAR VM66534, z/VM V7.2 supports host crypto-interruption exploitation for APVIRT cryptographic guests. The host is not required to poll cryptographic resources for replies that are ready to be delivered to the guest.

The following topics are updated:

- [“DETACH CRYPTO” on page 286](#)
- [“QUERY CRYPTO” on page 853](#)
- [“SET CRYPTO” on page 1476](#)

## Miscellaneous updates for May 2022

References to IBM Z® Application Assist Processor (zAAP), which is not supported on IBM z13 and later models, are removed:

- [“DEFINE CPU” on page 152](#)
- [“INDICATE LOAD” on page 535](#)
- [“INDICATE USER” on page 555](#)
- [“QUERY PROCESSORS” on page 1092](#)
- [“QUERY SHARE” on page 1161](#)
- [“QUERY SRM” on page 1173](#)
- [“QUERY VIRTUAL CPUS” on page 1282](#)
- [“SET SHARE” on page 1691](#)
- [“SET SRM” on page 1704](#)

## SC24-6268-10, z/VM 7.2 (December 2021)

---

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.2.

### [VM66560] Preserve Partial Dump After Initial Program Load (IPL)

z/VM APAR VM66560 preserves partial dumps that are caused when an IPL of the partition ends the dump process before the dump completes. Currently, partial dumps that are ended by an IPL of the partition are not preserved after the IPL.

The following topics are updated:

- [“DUMpload” on page 2180](#)
- [“DUMPLD2” on page 2184](#)

### [VM66557] VSwitch Bridge Port Enhancements

With the PTF for APAR VM66557, z/VM 7.2 adds a new NICDISTRIBUTION option to the VSwitch HiperSockets Bridge. When activated, the option enables the Bridge to distinguish and manage separately the traffic that is generated by various HiperSockets connections that are on the same HiperSockets CHPID. Traffic that exits the Bridge Port to an OSA link aggregation group is more evenly distributed across the entire port group. Activation of the new option also enables the VSwitch to extract IPv4 and IPv6 address assignments for display in the QUERY VSWITCH command, monitor records, and DIAGNOSE code X'26C'.

The following CP commands are updated:

- [“QUERY VSWITCH” on page 1351](#)
- [“SET VSWITCH” on page 1757](#)

## SC24-6268-09, z/VM 7.2 (August 2021)

---

This edition includes terminology, maintenance, and editorial changes.

Information about the following CP command is updated:

- [“QUERY PROCESSORS”](#) on page 1092. The explanation of the CORE element of the response is updated to indicate that the value is the ID for the logical core (not the ID of a physical core). "Real processor address" is replaced by "logical processor number".

## SC24-6268-09, z/VM 7.2 (July 2021)

---

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.2.

### **[VM66271] Dynamic Memory Downgrade (DMD) Enablement**

With the PTF for APAR VM66271, z/VM 7.2 provides support for the dynamic downgrading of memory. *Dynamic memory downgrade (DMD)*, also known as *memory reclamation* or *storage reclamation*, extends the real storage dynamic management characteristics of z/VM to include removing up to 50% of the real storage from a running z/VM system.

This support allows a system administrator to take real memory offline from a z/VM partition, making it available to other partitions in the central processor complex (CPC). The removal is dynamic; no re-IPLing of the z/VM image is required to accomplish the change in the memory configuration. Only memory that is configured as reconfigurable can be removed without an IPL.

Memory reclamation by a first-level z/VM system can be performed on the IBM z14 (or later) family of servers (or equivalent).

The following CP commands are updated:

- [“SET RESERVED”](#) on page 1674
- [“SET STORAGE”](#) on page 1722

### **[VM66485] IPv6 Layer 2 Query Support**

With the PTF for APAR VM66485, z/VM 7.2 provides support to collect and report IPv6 addresses for Layer 2 VSwitches. IPv6 addresses for Layer 2 VSwitches are displayed by the QUERY VSWITCH and QUERY VIRTUAL NIC commands and by DIAGNOSE code X'26C'. In addition, the VSwitch Unicast IP table structure is changed from a linked list to a hash table to allow for more efficient table searches.

The following CP commands are updated:

- [“QUERY LAN”](#) on page 966
- [“QUERY VIRTUAL NIC”](#) on page 1310
- [“QUERY VSWITCH”](#) on page 1351

### **[VM66496] Improved Live Guest Relocation among Crypto Environments with Mixed Crypto Adapter Types**

Live Guest Relocation (LGR) for APVIRT crypto users is supported even when the type of the shared crypto resource on the source system does not match the type on the target system. However, the crypto adapter mode (CCA or accelerator) on the source system must match the mode on the target system.

The QUERY VIRTUAL CRYPTO command on any system in a relocation domain reports the lowest type of crypto express adapter that is available in the shared pools of all systems in a relocation domain. Only an adapter in the user's relocation domain that is compatible with the same shared crypto mode as the current system is displayed.

APVIRT crypto users can issue the QUERY VIRTUAL CRYPTO command to determine the level of functions that can be used to enable relocation between systems in the relocation domain without using the FORCE ARCHITECTURE option.

The following CP commands are updated:

- [“QUERY VIRTUAL CRYPTO”](#) on page 1284

- [“VMRELOCATE” on page 2083](#)

## **[VM66507] EDEVICE Path Management**

With the PTF for APAR VM66507, z/VM improves the following aspects of EDEVICE path management:

- Path selection
- Path state tracking
- Path health checking

Updates to the following commands enable the user to display and clear detailed information associated with individual path activity:

- [“QUERY EDEVICE” on page 898](#)
- [“SET EDEVICE” on page 1504](#)

## **Miscellaneous updates for July 2021**

Information about the following CP command is updated:

- [“QUERY STORAGE/STORE” on page 1186](#)

The following command topics are updated to clarify the interaction of several shutdown-related commands and statements and the effects on shutdown time intervals:

- [“SET SIGNAL” on page 1701](#)
- [“SET SHUTDOWNTIME” on page 1698](#)
- [“SHUTDOWN” on page 1789](#)

## **SC24-6268-08, z/VM 7.2 (March 2021)**

---

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.2.

### **[VM66430] Fast Dump Distiller**

With the PTF for APAR VM66430, clients can distill a hard abend or snap dump to create an abridged dump, which can be submitted to IBM customer support for diagnosis. The abridged dump alleviates the problems and delays that can occur when you submit a large hard abend dump or snap dump to IBM. The original dump remains available as a spool file and can be loaded into a CMS file for transmission to IBM later, if necessary.

The following CP utility is updated: [“DUMpload” on page 2180](#).

### **[VM66173] 4 TB Real Memory Support**

z/VM APAR VM66173 delivers support for up to 4 TB of real memory, allowing z/VM systems to address a full 4 TB of first-level (real) memory, doubling the previous supported limit of 2 TB. With advanced memory management capabilities available in the z/VM product, clients now have the ability to run workloads that exceed 4 TB of virtual memory across all hosted guest systems, depending on workload characteristics. In conjunction with z/VM support for 80 processors, IBM Z and LinuxOne servers can now host even more work in a single z/VM partition, or across multiple z/VM partitions on one system. APAR VM66173 also delivers various system command updates, such as automatic STANDBY memory for guests and an enhancement to the CP DEFINE STORAGE command.

With automatic STANDBY memory for guests, a system administrator can code a more generic DEFINE STORAGE command that does not need to be updated every time the guest's directory entry storage size changes.

The following CP commands are updated:

- [“ATTACH” on page 30](#)
- [“DEFINE STORAGE” on page 213](#)
- [“QUERY FRAMES” on page 925](#)
- [“QUERY PAGING” on page 1060](#)
- [“QUERY STORAGE/STORE” on page 1186](#)
- [“QUERY SXSSTORAGE” on page 1199](#)
- [“SET AGELIST” on page 1418](#)
- [“SET PAGING” on page 1596](#)
- [“SET STORAGE” on page 1722](#)

The following CP utility is updated: [“CPSYNTAX” on page 2143](#)

## **[VM66201] z/Architecture Extended Configuration (z/XC) support**

With the PTFs for APARs VM66201 (CP) and VM66425 (CMS), z/Architecture® Extended Configuration (z/XC) support is provided. CMS applications that run in z/Architecture can use multiple address spaces. A z/XC guest can use VM data spaces with z/Architecture in the same way that an ESA/XC guest can use VM data spaces with Enterprise Systems Architecture. z/Architecture CMS (z/CMS) can use VM data spaces to access Shared File System (SFS) Directory Control (DIRCONTROL) directories. Programs can use z/Architecture instructions and registers (within the limits of z/CMS support) and can use VM data spaces in the same CMS session. For more information, see [z/VM: z/Architecture Extended Configuration \(z/XC\) Principles of Operation](#).

Information about the following CP commands and utilities is updated:

- [“DISPLAY” on page 331](#)
- [“DISPLAY \(Guest Storage - ESA/XC, z/XC\)” on page 337](#)
- [“DISPLAY PSW” on page 373](#)
- [“DISPLAY PSWG” on page 376](#)
- [“DISPLAY \(Registers\)” on page 378](#)
- [“DUMP” on page 395](#)
- [“DUMP \(Guest Storage - ESA/XC, z/XC\)” on page 401](#)
- [“DUMP PSW” on page 434](#)
- [“DUMP PSWG” on page 437](#)
- [“DUMP \(Registers\)” on page 439](#)
- [“DEFINE DIAGNOSE” on page 172](#)
- [“DEFSYS” on page 253](#)
- [“IPL” on page 566](#)
- [“LOGON” on page 662](#)
- [“QUERY SET” on page 1156](#)
- [“SAVESYS” on page 1397](#)
- [“SET MACHINE” on page 1555](#)
- [“SET PAGEX” on page 1594](#)
- [“SET VTOD” on page 1781](#)
- [“SET 370ACCOM” on page 1787](#)
- [“STORE” on page 1879](#)
- [“STORE \(Guest Storage - ESA/XC, z/XC\)” on page 1881](#)
- [“STORE PSW” on page 1901](#)
- [“STORE PSWA” on page 1903](#)

- [“STORE PSWG” on page 1904](#)
- [“STORE \(Registers\)” on page 1905](#)
- [“TRACE” on page 1946](#)
- [“TRACE: Options” on page 1949](#)
- [“TRACE: Command Responses” on page 1954](#)
- [“TRACE GG” on page 1978](#)
- [“TRACE STORE \(ESA/390, z/Architecture\)” on page 1991](#)
- [“TRACE STORE \(ESA/XC, z/XC\)” on page 1995](#)
- [“VMDUMP” on page 2072](#)
- [“XAUTOLOG” on page 2100](#)
- [“SDINST” on page 2235](#)

## Miscellaneous updates for March 2021

The following elements are changed or deleted in this and later versions of the publication.

### **[“CPFMTXA” on page 2124](#)**

In the usage notes, consideration is added for the Stand-Alone Program Loader (SAPL) on an FBA device.

### **[“DEFINE MDISK” on page 188](#)**

The explanation of the **start** operand adds consideration of CKD and FBA DASD extents.

### **[“LOGON” on page 662](#)**

The following note about the STORAGE operand is obsolete and is deleted:

- An XC virtual machine can address up to 2047 MB of storage in its base address space.

In table [Maximum Input Values for Storage Units](#), the value for kilobytes is changed to 99999999, and the value for terabytes is changed to 16777216. The previously published values were incorrect.

### **[“SET VTOD” on page 1781](#)**

Message HCP6152E is deleted. The following usage note is obsolete and is deleted:

- The virtual machine whose TOD clock you want to change may not be in ESA/XC mode.

### **[“VMDUMP” on page 2072](#)**

The following usage notes are obsolete and are deleted:

- Access registers are dumped only for XA, ESA, XC, and z/Architecture virtual machines.
- VMDUMP dumps of z/Architecture guests on z/VM version 5.2 are not compatible with VMDUMP dumps of z/Architecture guests of previous releases.

### **[“XAUTOLOG” on page 2100](#)**

The following note about the STORAGE operand is obsolete and is deleted:

- An XC virtual machine can address up to 2047 MB of storage in its base address space.

## SC24-6268-07, z/VM 7.2 (December 2020)

---

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.2.

### **[VM66469] CP QUERY DEVICES Command**

The CP QUERY DEVICES command displays device setting information. The command is an extension of the existing support that is intended to simplify configuration management and problem determination. See [“QUERY DEVICES” on page 883](#).

## **[VM66420] Optional LUN Specification on SET EDEVICE Command**

The LUN operand is required when you define the first path to an EDEVICE. After the path is established, the LUN operand is optional when you define subsequent paths. See [“SET EDEVICE” on page 1504](#).

## **[VM66421] Guest HyperPAV Support for 1-END Minidisks**

Allows clients to define and associate virtual HyperPAV alias devices with 1-END base minidisks. The DEFINE HYPERPAVALIAS command does not change, but the requirement to use real cylinder 0 is eliminated. See [“DEFINE HYPERPAVALIAS” on page 180](#).

## **[VM66433] FlashCopy PRESERVEMIRROR Support**

Provides new operands to z/VM FlashCopy® commands to allow FlashCopy relationships between PPRC primary devices without affecting the PPRC status of the target devices. The new operands avoid PPRC suspend and duplex pending conditions that are associated with the target device of the FlashCopy. See [“FLASHCOPY” on page 471](#) and [“FLASHCOPY ESTABLISH” on page 483](#).

## **[VM66439] CP New Feature Interrogation API (New function indicator)**

Provides an interface for interrogating whether a named capability is present in the running CP system. This enhancement simplifies scripting and automation written by customers and vendors sensitive to the function level of CP. It facilitates adaptation of vendor products to varying CP code levels. See [“QUERY VARIABLE” on page 1267](#).

# **SC24-6268-06, z/VM 7.2 (September 2020)**

---

This edition includes changes to support the general availability of z/VM 7.2.

## **Adjunct Virtual Machine Support**

z/VM 7.2 has been updated to support adjunct virtual machines. An adjunct is a second virtual machine configuration running under your user ID, distinct from the configuration created when your user ID is logged on or autologged (called the principal configuration). Each configuration has its own virtual CPU(s), virtual storage (memory), and virtual devices, and can IPL a different instance of an operating system. The adjunct is intended to run CMS, to give you access to CMS files and tools, for example, to aid in debugging the operating system in the principal configuration while keeping the principal's operating system intact.

The following CP commands have been added:

- [“ADJUNCT” on page 18](#)
- [“QUERY ADJUNCT” on page 756](#)
- [“SET ADJUNCTS” on page 1417](#)

The following CP commands have been updated:

- [“CPACCESS” on page 89](#)
- [“CPU” on page 107](#)
- [“DEFINE” on page 122](#)
- [“DEFINE MDISK” on page 188](#)
- [“LINK” on page 578](#)
- [“LOGON” on page 662](#)
- [“QUERY LINKS” on page 979](#)
- [“QUERY VIRTUAL CPUS” on page 1282](#)

## MSS Multi-Target PPRC Exploitation

z/VM now supports up to four subchannel sets that reflect the actual hardware configuration.

The following CP commands have been updated:

- [“DELETE EDEVICE” on page 268](#)
- [“DELETE RDEVICE” on page 274](#)
- [“HYPERSWAP” on page 525](#)
- [“LOCATE RDEV” on page 618](#)
- [“QUERY CHPID” on page 791](#)
- [“QUERY DASD” on page 867](#)
- [“QUERY EDEVICE” on page 898](#)
- [“QUERY EQID” on page 905](#)
- [“QUERY HYPERSWAP” on page 941](#)
- [“QUERY MSS” on page 1025](#)
- [“QUERY PATHS” on page 1064](#)
- [“QUERY \(Real Device\)” on page 1114](#)
- [“SET DEVICES” on page 1490](#)
- [“SET EDEVICE” on page 1504](#)
- [“SET RDEVICE DASD” on page 1641](#)
- [“VARY PATH” on page 2053](#)
- [“VARY \(Real Device\)” on page 2063](#)
- [“VARY SUBCHANNEL” on page 2067](#)

The following CP utilities have been updated:

- [“CPSYNTAX” on page 2143](#)
- [“IOEXPLOR” on page 2205](#)
- [“SALIPL” on page 2222](#)

## Removal of PAGING63 IPL Parameters

The PAGING63 IPL parameter is removed because this IPL parameter blocks use of newer paging technologies (EAV, Encrypted, HPF Paging, and HyperPAV, for example) and has not been recommended for use since z/VM 6.4.

The following CP commands have been updated:

- [“QUERY PAGING” on page 1060](#)
- [“SET ENCRYPT” on page 1513](#)
- [“SET IPLPARMS” on page 1530](#)
- [“SET PAGING” on page 1596](#)

## Miscellaneous updates for September 2020

The following CP utility has been updated:

- [“MONWRITE” on page 2212](#)



---

## Chapter 1. Introduction

In this introduction, the term *command* is used generically; it refers to both CP commands and CP utilities.

z/VM uses command languages to correspond to the two environments it creates, control program and virtual machine.

Use the control program (CP) command language when:

- You are a z/VM system operator and you want to control the resources of the real machine located in your computer room.
- You are a virtual machine user and you want to control your virtual machine's configuration and environment.

Use a virtual machine command language when:

- You are communicating with the operating system you loaded into your virtual machine.

**To perform production or test work**, load your virtual machine with one of the operating systems supported by the z/VM system. Your virtual machine command language is the command language of the operating system you load. This command language is described in the library that documents that particular operating system.

**To perform service, installation, and maintenance tasks, along with editing and text creation, communicating with others, and problem solving**, load your virtual machine with the conversational monitor system (CMS). CMS is a single user, conversational operating system.

---

## Using Commands

You can use CP commands whenever:

- Your virtual machine is in the control program (CP) command environment. Your virtual machine is in the CP environment when you log on to z/VM and CP READ is displayed in the lower right-hand corner of the screen. On a line-mode ASCII device, there is no status area to display CP READ, so CP is displayed in the output area.
- You press the break key while in full-screen mode before entering a command. (The break key may be PA1, the VM default break key, or another key that you have defined as the break key using the TERMINAL BRKKEY command. Also, the break key may be totally disabled by some application programs or when in the protected application environment.)
- You are in a virtual machine command environment, not running in full-screen mode, and enter the #CP command (and # is your logical linend character).
- You are in the CMS virtual machine environment and enter the CP command.
- You are in the CMS virtual machine environment and have the IMPCP function set ON.

To determine the current command environment on a 3270 device, look at the status area in the lower right-hand corner of the display screen. CP READ indicates the CP environment, and VM READ indicates the virtual machine environment. If you are running CMS in your virtual machine, VM READ indicates the CMS environment. When RUNNING appears in the status area, enter a null input line to determine your environment. (To enter a null line, press the ENTER key but do not enter any data.) When you enter the null line, the status area displays either CP READ or VM READ. Also, if you are in a read state, in either the CP command environment (with RUN set OFF) or the CMS command environment, and you enter a null line, the system responds with the name of your command environment: CP or CMS in the system output area.

To determine the command environment for a line-mode device, a null entry (carriage return with no data) yields CP or CMS in the system output area.

## Introduction

You enter CP commands using any combination of uppercase and lowercase letters. When you have typed the command and its operands, press the ENTER key to process the command.

You may enter commands from CMS EXEC procedures or applications. See *z/VM: System Operation* and *z/VM: REXX/VM User's Guide*, and the description of `DIAGNOSE X'08'` in *z/VM: CP Programming Services*, for information on the programming interfaces for simulating the virtual console.

When simulating a virtual console, keep the following considerations in mind:

- Textual output from commands may change from release to release. In particular, command output that affects the way you use a terminal may be rearranged. NLS support may make the output available in multiple languages. Textual output is changed only when necessary.
- Return codes remain relatively stable from release to release.

## Command Structure

A VM command line consists of a command name, usually followed by one or more positional operands. The general form for the CP command line is:

command name	operands
--------------	----------

You must use one or more blanks to separate each entry in the command line unless otherwise indicated.

### The Command Name

The command name is an alphanumeric symbol, not more than 12 characters long. For example, DISCONNECT.

In general, command names are verbs that describe the function you want the system to perform. For example, if you want to attach a real device to your virtual machine configuration, enter the ATTACH command.

### The Command Operands

Command operands are keywords and positional symbols, usually no longer than eight alphanumeric characters.

You use operands to specify the information on which the system will operate when it performs the command function. For example, you use the TERMINAL operand in conjunction with the CP QUERY command to find out what terminal options are in effect for your virtual machine.

Some commands require no operands; others require several. You can find a description of each command with all of its operand requirements in [Chapter 2, "CP Commands,"](#) on page 11 or [Chapter 3, "CP Utilities,"](#) on page 2113.

Unless otherwise indicated, you must enter the operands you select in the order in which they appear in the command format.

### How to Enter Comments in the CP Command Language

You can use the CP asterisk (\*) command to annotate your display screen and console spool file. (Within this file are your input to the system and the system's response to you.) Refer to ["ASTERISK \(\\*\)"](#) on page 12 for complete details.

### Character Sets

You can enter CP commands using a combination of characters from 6 different character sets. The contents of each of these sets is summarized in [Table 1](#) on page 3.

Character Set	Names	Symbols
Separator	Blank	
National	Dollar Sign Pound Sign At Sign	\$ # @
Alphabetic	Uppercase Lowercase	A through Z a through z
Numeric	Numeric	0 through 9
Alphanumeric	National Alphabetic  Numeric	\$, #, @ A through Z a through z 0 through 9
Special		All other characters

## Using Truncations and Abbreviations

To make entering commands on the keyboard more convenient, VM lets you enter many commands and operands in a shortened form. This shortened form can be either a truncation or an abbreviation.

You truncate a command or operand name by dropping one or more letters from the end of the name. The syntax box for each command shows the truncations you can use for each command and its operands. If a letter appears in uppercase in the syntax box, you must enter it when you enter the command from your display; if a letter is in lowercase, you may omit it when you enter the command.

For example, the syntax box for the QUERY command shows the command as:

```
Query
```

This means that you can enter the QUERY command in any of the following forms:

```
query
quer
que
qu
q
```

The minimum acceptable truncation is Q, but CP will accept any of these forms of the command.

Abbreviations are also shorter forms of commands and operands, but unlike truncations, are not formed by simply dropping letters from the end of the command name.

Command syntax boxes show acceptable abbreviations for command names. The abbreviations appear below the full name of the command. Operand abbreviations are listed in the operand descriptions following the syntax box.

For example, the syntax box for the MESSAGE command shows:

```
Message
Msg
```

This means that you can:

- Truncate the command to a minimum of “M”
- Use the abbreviation “MSG”

## Introduction

- Truncate the abbreviation to “MS” or “M”.

Thus, CP accepts all of the following forms of the MESSAGE command:

```
message
messag
messa      OR      msg
mess       ms
mes        m
me
m
```

## Syntax, Message, and Response Conventions

The following topics provide information on the conventions used in syntax diagrams and in examples of messages and responses.

### How to Read Syntax Diagrams

Special diagrams (often called *railroad tracks*) are used to show the syntax of external interfaces.

To read a syntax diagram, follow the path of the line. Read from left to right and top to bottom.

- The  $\blacktriangleright$ — symbol indicates the beginning of the syntax diagram.
- The — $\blacktriangleright$  symbol, at the end of a line, indicates that the syntax diagram is continued on the next line.
- The  $\blacktriangleright$ — symbol, at the beginning of a line, indicates that the syntax diagram is continued from the previous line.
- The — $\blacktriangleright$ ◀ symbol indicates the end of the syntax diagram.

Within the syntax diagram, items on the line are required, items below the line are optional, and items above the line are defaults. See the examples in [Table 2 on page 4](#).

Syntax Diagram Convention	Example
<b>Keywords and Constants</b> A keyword or constant appears in uppercase letters. In this example, you must specify the item KEYWORD as shown. In most cases, you can specify a keyword or constant in uppercase letters, lowercase letters, or any combination. However, some applications may have additional conventions for using all-uppercase or all-lowercase.	$\blacktriangleright$ KEYWORD $\blacktriangleleft$
<b>Abbreviations</b> Uppercase letters denote the shortest acceptable abbreviation of an item, and lowercase letters denote the part that can be omitted. If an item appears entirely in uppercase letters, it cannot be abbreviated. In this example, you can specify KEYWO, KEYWOR, or KEYWORD.	$\blacktriangleright$ KEYWOrd $\blacktriangleleft$

Table 2. Examples of Syntax Diagram Conventions (continued)

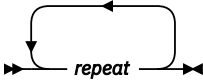
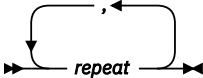
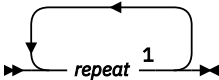
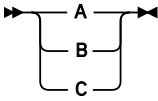
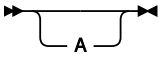
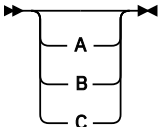
Syntax Diagram Convention	Example
<p><b>Symbols</b></p> <p>You must specify these symbols exactly as they appear in the syntax diagram.</p>	<p>* Asterisk</p> <p>:</p> <p>Colon</p> <p>,</p> <p>Comma</p> <p>=</p> <p>Equal Sign</p> <p>-</p> <p>Hyphen</p> <p>()</p> <p>Parentheses</p> <p>.</p> <p>Period</p>
<p><b>Variables</b></p> <p>A variable appears in highlighted lowercase, usually italics.</p> <p>In this example, <i>var_name</i> represents a variable that you must specify following KEYWORD.</p>	<p>▶▶ KEYWORD — <i>var_name</i> ▶▶</p>
<p><b>Repetitions</b></p> <p>An arrow returning to the left means that the item can be repeated.</p> <p>A character within the arrow means that you must separate each repetition of the item with that character.</p> <p>A number (1) by the arrow references a syntax note at the bottom of the diagram. The syntax note tells you how many times the item can be repeated.</p> <p>Syntax notes may also be used to explain other special aspects of the syntax.</p>	   <p>Notes:</p> <p><sup>1</sup> Specify <i>repeat</i> up to 5 times.</p>
<p><b>Required Item or Choice</b></p> <p>When an item is on the line, it is required. In this example, you must specify A.</p> <p>When two or more items are in a stack and one of them is on the line, you must specify one item. In this example, you must choose A, B, or C.</p>	<p>▶▶ A ▶▶</p> 
<p><b>Optional Item or Choice</b></p> <p>When an item is below the line, it is optional. In this example, you can choose A or nothing at all.</p> <p>When two or more items are in a stack below the line, all of them are optional. In this example, you can choose A, B, C, or nothing at all.</p>	 

Table 2. Examples of Syntax Diagram Conventions (continued)	
Syntax Diagram Convention	Example
<p><b>Defaults</b></p> <p>When an item is above the line, it is the default. The system will use the default unless you override it. You can override the default by specifying an option from the stack below the line.</p> <p>In this example, A is the default. You can override A by choosing B or C.</p>	
<p><b>Repeatable Choice</b></p> <p>A stack of items followed by an arrow returning to the left means that you can select more than one item or, in some cases, repeat a single item.</p> <p>In this example, you can choose any combination of A, B, or C.</p>	
<p><b>Syntax Fragment</b></p> <p>Some diagrams, because of their length, must fragment the syntax. The fragment name appears between vertical bars in the diagram. The expanded fragment appears in the diagram after a heading with the same fragment name.</p> <p>In this example, the fragment is named "A Fragment."</p>	

## Examples of Messages and Responses

Although most examples of messages and responses are shown exactly as they would appear, some content might depend on the specific situation. The following notation is used to show variable, optional, or alternative content:

**xxx**

Highlighted text (usually italics) indicates a variable that represents the data that will be displayed.

**[ ]**

Brackets enclose optional text that might be displayed.

**{ }**

Braces enclose alternative versions of text, one of which will be displayed.

**|**

The vertical bar separates items within brackets or braces.

**...**

The ellipsis indicates that the preceding item might be repeated. A vertical ellipsis indicates that the preceding line, or a variation of that line, might be repeated.

## Privilege Classes

The set of CP commands that you can use depends upon the privilege class or classes that have been assigned to you.

CP commands are divided into eight groups, each represented by a privilege class. The privilege class indicates the type of user from whom the system accepts commands.

In general, the system programmer who creates your system directory assigns you one or more privilege classes as part of your entry in the directory.

Privilege classes are denoted by the letters A through Z, the numbers 1 through 6, or the word “Any.” These classes, and the type of user who can use the commands belonging to each privilege class set, are summarized in [Table 3 on page 7](#). Classes I through Z and numbers 1 through 6 are reserved so that your installation can define them to suit its needs.

If you are authorized for some options of a command but not for others and you enter an unauthorized option, your command will be interpreted in the formats for which you are authorized. This may cause the command to either:

- execute in an unexpected fashion, or
- reflect an unexpected message.

Users whose password is NOLOG have no privilege class and can only receive spooled output as punched cards or printed forms. The NOLOG assignment is controlled by directory statements; see [IDENTITY Directory Statement in z/VM: CP Planning and Administration](#) for more information about NOLOG users.

Class	User and Function
A	<i>System Operator:</i> The class A user controls the z/VM system. The system operator is responsible for the availability of the z/VM system and its resources. In addition, the system operator controls system accounting, broadcast messages, virtual machine performance options, and other options that affect the overall performance of z/VM. <b>Note:</b> The class A user who is automatically logged on during CP initialization is designated as the primary system operator.
B	<i>System Resource Operator:</i> The class B user controls all the real resources of the z/VM system, except those controlled by the system operator and the spooling operator.
C	<i>System Programmer:</i> The class C user updates or changes system-wide parameters of the z/VM system.
D	<i>Spooling Operator:</i> The class D user controls spool files and the system's real reader, printer, and punch equipment allocated to spooling use.
E	<i>System Analyst:</i> The class E user examines and saves system operation data in specified z/VM storage areas.
F	<i>Service Representative:</i> The class F user obtains, and examines in detail, data about input and output devices connected to the z/VM system. This privilege class is reserved for IBM use only.
G	<i>General User:</i> The class G user controls functions associated with a particular virtual machine.
Any	Commands belonging to class “Any” are available to any user, regardless of the user's privilege class. These commands are primarily those used to gain access to, or relinquish access from, the z/VM system.
H	Reserved for IBM use.
I - Z 1 - 6	These classes are reserved for redefinition by each installation for its own use (using MODIFY statements or commands).

## Delayed Responses to CP Commands

Many CP commands associated with DASD subsystems result in a channel program being issued to a device or subsystem. In most cases, these operations complete in a very short time, and the command

## Introduction

response is presented to the operator immediately. However, for IBM DASD subsystems, some of these operations do not complete immediately. The command responses associated with such operations are called delayed responses.

The CP commands that may result in delayed responses are:

- COMMIT
- FLASHCOPY
- SET CACHE ON SUBSYSTEM
- SET CACHE OFF SUBSYSTEM
- SET CACHE OFF DEVICE
- SET CACHEFW OFF
- SET DASDFW OFF
- SET NVS ON
- SET NVS OFF

The responses to these commands are displayed in the following manner.

If the command completes immediately, the following response is displayed:

```
Normal command response.
```

If the command does not complete immediately, the following response is displayed:

```
Command started:  command details
```

If the command completes successfully, the following response is displayed:

```
Normal command response.
```

If errors occur during the asynchronous processing of the command that prevent the successful completion of the command, the following response is displayed:

```
Command failed:  command details
```

Under certain circumstances, it is possible that a cache storage subsystem will not notify CP when it has completed a command. When this occurs, the usual completion responses cannot be generated. Under these circumstances, command processing is ended after displaying the following response:

```
Command results lost:  command details
```

If the originator of the command is not the system operator, the following response may be issued to the system operator when the command completes.

```
Normal command response by userid.
```

The user may display the status of commands that the user has issued that have not yet completed by issuing the QUERY PENDING COMMANDS commands. QUERY PENDING COMMANDS causes the following response to be displayed for each pending CP command:

```
Command pending:  command details
```

The user may display the status of ALL commands that are pending for a device by issuing the QUERY PENDING COMMANDS ALLUSERS command. The ALLUSERS option causes one of the following responses to be displayed for each pending CP command:

```
Command pending for userid: command details  
OR  
Command pending: command details
```



**Normal command response.**

is the usual response associated with the successful completion of this command.

**userid**

is the ID of the user who issued the command if the command was not issued by the issuer of the QUERY PENDING COMMANDS ALLUSERS command.

**command details**

is the command and options with which this delayed response is associated.

**Example**

Assume that a class B user, DASDOPER, wants to deactivate the DASD fast write function for devices 190 through 194 by issuing SET DASDFW DEVICE OFF 190-194. Since SET DASDFW OFF may produce delayed responses, all of the following results are possible. Assume that the devices respond as follows:

- 190 - responds immediately as successful
- 191 - delayed response buffer indicates completed successfully
- 192 - delayed response buffer indicates completed with errors
- 193 - delayed response buffer indicates command failed
- 194 - polling detects that the response was lost.

**Note:** The operator is informed of all successful operations.

**Command Issuer's Console (DASDOPER)**

```
SET DASDFW DEVICE OFF 190-194
190 DASD fast write function has been deactivated.
Command started: SET DASDFW OFF 191.
Command started: SET DASDFW OFF 192.
Command started: SET DASDFW OFF 193.
Command started: SET DASDFW OFF 194.
:
:
QUERY PENDING COMMANDS 190-194
No pending delayed responses exist for 190.
Command pending: SET DASDFW OFF 191.
Command pending: SET DASDFW OFF 192.
Command pending: SET DASDFW OFF 193.
Command pending: SET DASDFW OFF 194.
:
:
191 DASD fast write function has been deactivated.
:
:
192 DASD fast write function has been deactivated.
:
:
Command failed: SET DASDFW OFF 193.
:
:
Command results lost: SET DASDFW OFF 194.
```

**System Operator's Console**

```
190 DASD fast write function was deactivated by DASDOPER.
:
:
QUERY PENDING COMMANDS ALLUSERS 190-194
No pending delayed responses exist for 190.
Command pending for DASDOPER: SET DASDFW OFF 191.
Command pending for DASDOPER: SET DASDFW OFF 192.
Command pending for DASDOPER: SET DASDFW OFF 193.
Command pending for DASDOPER: SET DASDFW OFF 194.
:
:
191 DASD fast write function was deactivated by DASDOPER.
:
:
192 DASD fast write function was deactivated by DASDOPER.
```

# Using the Online HELP Facility

---

You can receive online information about the commands described in this book using the z/VM\* HELP Facility. For example, to display a menu of CP commands, enter:

```
help cp menu
```

To display information about a specific CP command (BEGIN in this example), enter:

```
help cp begin
```

You can also display information about a message by entering one of the following commands:

```
help msgid or help msg msgid
```

For example, to display information about message HCP002E, you can enter one of the following commands:

```
help hcp002e or help msg hcp002e
```

For more information about [Using the HELP Facility](#), see *z/VM: CMS User's Guide*. To display the main HELP Task Menu, enter:

```
help
```

For more information about the [HELP](#) command, see *z/VM: CMS Commands and Utilities Reference* or enter:

```
help cms help
```

---

## Chapter 2. CP Commands

This section contains reference information on the CP commands. These commands are listed in alphabetical order except for the \* and #CP commands, which appear first.

Each command description is presented in the following format (some sections might not be included):

- **Name:** Identifies the name of the command.
- **Format:** Shows the syntax of the command with all the possible operands and options you can use.
- **Authorization:** Specifies the required CP privilege class, which identifies the users able to issue the command, unless user class modification is in use.
- **Purpose:** States what the command is used for.
- **Operands and Options:** Defines the function of each operand and option and any values you can include.
- **Usage Notes:** Identifies and describes special situations and other considerations that may affect your use of the command.
- **Examples:** Provides one or more examples to show how the command is commonly used.
- **Responses:** Describes the responses you might receive from the command on your display device. Responses are normal operational output; they tell you about the execution and effect of the command. Unlike system messages, command responses are not prefixed with an identifying number and are not contained in *z/VM: CP Messages and Codes*.
- **Messages:** Lists the messages issued by the command. Messages not unique to the command might also be issued. Each message is prefixed with an identifying number. For more information on the messages, including suggested actions, see *z/VM: CP Messages and Codes*.

## ASTERISK (\*)

---



### Authorization

Privilege Class: Any

### Purpose

Use ASTERISK (\*) to annotate your console sheet or display screen.

### Usage Notes

1. The control program (CP) treats any statement prefixed by an asterisk (\*) as a comment line.
2. If you start the console spooling function for your virtual machine, your commentary appears in your console spool file.

## #CP



### Authorization

Privilege Class: Any

### Purpose

Use #CP to process one or more control program (CP) commands from a guest environment.

### Operands

#### *command*

identifies the name and operands of the CP command or commands you want to enter. You must precede the first command with at least one blank. If you are specifying multiple CP commands, separate them with your logical-line-end symbol. (See Usage Note “4” on page 13.)

### Usage Notes

1. The pound sign (#) shown above represents the logical-line-end symbol in effect for your virtual machine. If you have redefined the logical-line-end symbol, you must substitute your symbol for the pound sign (#).
2. The #CP command is a handy way to reduce the number of operations required to enter a CP command from a virtual console. For example, to display the number of files you have on your virtual machine, enter **cp query files**. This command is equivalent to the following operations:
  - a. Press PA1 to enter the CP command environment.
  - b. Enter QUERY FILES to display the number of files.
  - c. Enter BEGIN to exit the CP command environment.
3. Before you enter the #CP command, be sure that:
  - Your virtual machine is operating with SET LINEDIT on. (This is the default.)
  - The first three characters you enter in the command are #CP (uppercase or lowercase); the pound sign (#) represents your currently defined logical-line-end symbol.
  - At least one blank separates the #CP from the command itself.
4. You can enter multiple commands as operands of #CP command if you separate each command from the next with the logical-line-end symbol.
5. The #CP command has no meaning if you are running in full-screen mode.
6. Table 4 on page 13 summarizes the system response to various #CP commands:

Table 4. System Responses to Various #CP Commands

Command	System Action
#CP	Your virtual machine enters the CP environment. An exception to this is when the command is issued with the AT command. See usage note “7” on page 14 for details.
#CP QUERY FILES	The QUERY FILES command is executed.

---

*Table 4. System Responses to Various #CP Commands (continued)*

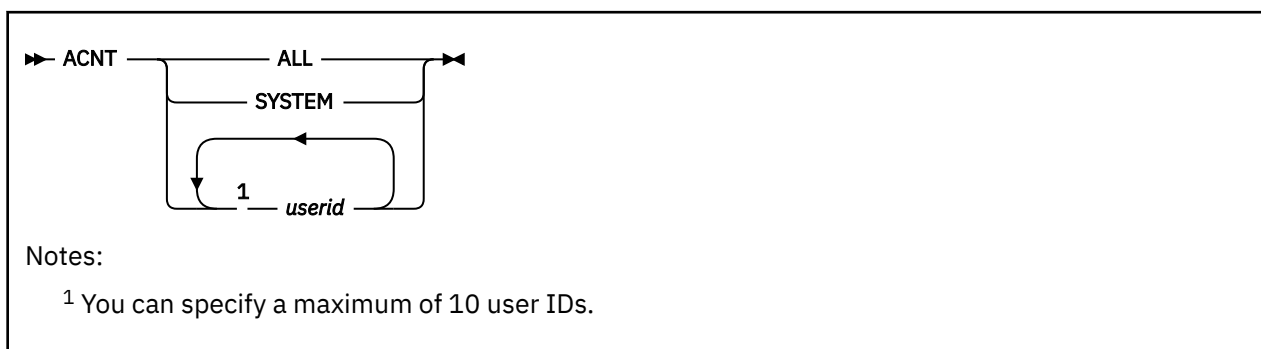
---

<b>Command</b>	<b>System Action</b>
#CP QUERY FILES#QUERY USERS	Two separate QUERY commands are executed.
#CP data entered	You enter the CP environment and CP interprets the "data entered" as a CP command; the command may or may not be valid.

---

7. The #CP command is rejected with message HCP026E if it is issued with the AT command. If preceded by your terminal escape character, it is rejected with message HCP1144E.
8. The CP command, when not preceded with the logical line-end character, and when issued with the AT command, will not cause any virtual machine on the target system to enter the CP environment. However, the CP command can still be used to prefix other CP commands when issued with the AT command.

## ACNT



### Authorization

Privilege Class: A

### Purpose

Use ACNT to do the following:

- Create accounting records for active users
- Reset accumulated accounting data fields that are reset to zero after a record is created.

### Operands

#### ALL

tells the system to create accounting records for all logged-on users and for the system.

#### SYSTEM

tells the system to account for all time not charged to a specific user.

#### *userid*

identifies the list of users for whom you want the system to create accounting records. You can specify up to 10 user IDs on the command input line.

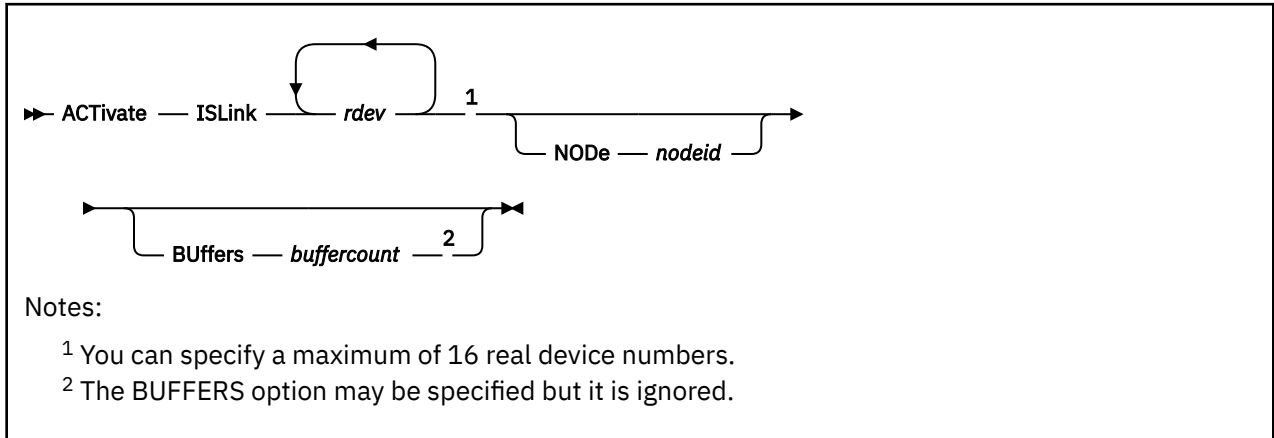
### Usage Notes

1. When you enter ACNT, all available types of accounting records are created for each user specified. For an explanation of individual accounting records, their contents, and how to set up a service virtual machine to collect them, refer to [Setting Up Service Virtual Machines in z/VM: CP Planning and Administration](#).
2. CP processes the list of user IDs in the order you enter them. If any user ID in the list is not valid, the creation of accounting records stops with the last valid user ID encountered.
3. To provide an ACNT command syntax compatible with VM/SP, z/VM accepts CLOSE as an option following ALL or ACNT (for example, ACNT ALL CLOSE or ACNT CLOSE). However, this option does not change the operation of the ACNT command.

### Messages

- HCP002E Invalid operand - *operand*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on

## ACTIVATE ISLINK



### Authorization

Privilege Class: B

### Purpose

Use the ACTIVATE ISLINK command to identify a communication link to ISFC.

### Operands

#### **rdev**

identifies one to sixteen real device numbers you want to use as a link.

When the first device is initialized for a new link, the ISLINK for that node is created automatically.

#### **NODe nodeid**

is the optional node (gateway) identifier for the ISFC link expected for each *rdev*. If not specified, it is determined by device initialization.

If a conflicting *nodeid* is specified for an already active device, an error message is issued.

#### **BUffers buffercount**

is ignored and included for compatibility only.

### Usage Notes

1. You may also need to define the real communication link to the system; see the [RDEVICE Statement](#) in *z/VM: CP Planning and Administration* for more information.
2. The system operator receives all informational and error messages, but not immediate command responses.
3. You can improve throughput across your ISFC link by activating more than one device on the same link. Note, however, that FICON® subchannels on the same channel path share some hardware resources. In general you will observe better data transfer rates if subchannels on a given link are configured to use different channel paths.
4. Avoid using different CHPID speeds in any one logical link, as that will cause the ISFC logical link to operate less efficiently.

### Responses

Response 1:



Link *rdev* activated.

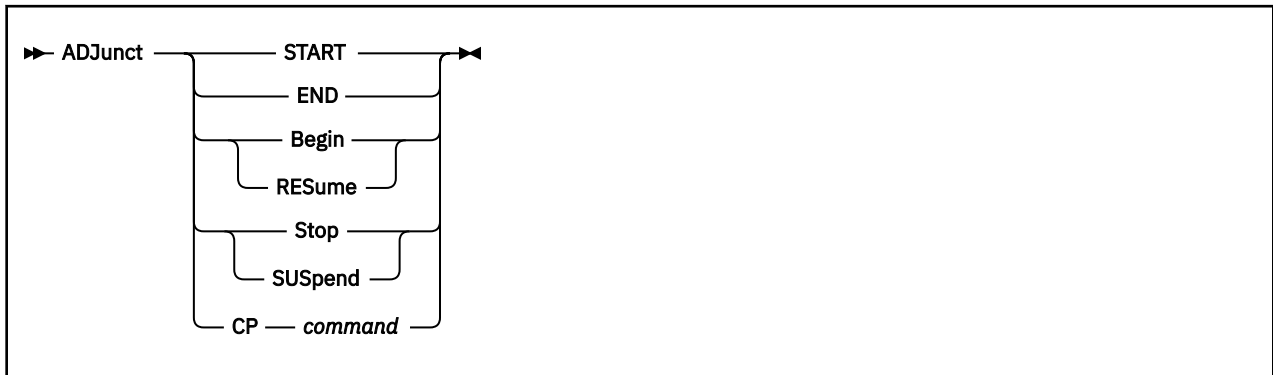
***rdev***

is the link address that has been activated.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP006E Invalid device type - *rdev*
- HCP026E Operand missing or invalid
- HCP040E Device *rdev* does not exist
- HCP046E Device *rdev* offline
- HCP049E Device *rdev* in use
- HCP2702I Link *linkname* came up
- HCP2704I Node *nodeid* added to collection
- HCP2706I Link *linkname* activated by user *userid*
- HCP2721E Link unit address *rdev* is a duplicate
- HCP2722I The BUFFERS option is ignored for ACTIVATE ISLINK.
- HCP2725E Inter-System Facility for Communication initialization failed: RC=*return-code*
- HCP2745E Link device *rdev* rejected. Defined node *nodeid1* conflicts with detected node *nodeid2*.
- HCP2750I ISFC ACTIVATE is deferred for offline device *devno*.
- HCP6005E Option BUFFERS is not supported by z/VM.

## ADJUNCT



### Authorization

Privilege Class: G

### Purpose

Use the ADJUNCT command to control the operation of an adjunct virtual machine.

An adjunct is a second virtual machine configuration running under your userid, distinct from the configuration created when your userid is logged on or autologged (called the principal configuration). Each configuration has its own virtual CPU(s), virtual storage (memory), and virtual devices, and can IPL a different instance of an operating system. The adjunct is intended to run CMS, to give you access to CMS files and tools, for example, to aid in debugging the operating system in the principal configuration while keeping the principal's operating system intact.

When both principal and adjunct configurations exist for a user, only one is runnable at a time. When the adjunct is started or resumed, the principal is suspended, and vice versa.

### Operands

#### START

The ADJUNCT START operation is analogous to LOGON for the primary configuration. It creates and IPLs an adjunct configuration, suspends execution of the principal configuration, switches the terminal session (if connected) from the virtual console of the principal to that of the adjunct, and transfers control to the adjunct configuration.

#### END

Logs off the adjunct, leaving the principal configuration active.

#### Begin

#### RESume

Suspends execution of the principal configuration, switches the virtual console to the adjunct, and transfers control to the adjunct configuration. The adjunct must already have been started. This operation is the adjunct analog to the CP BEGIN command for the principal configuration.

#### Stop

#### SUSpend

Puts the adjunct into a soft-stop state and transfers control back to the principal configuration. This operation is the adjunct analog to the CP STOP command for the principal configuration.

#### CP *command*

Permits a user or program in the principal configuration to direct commands to the adjunct configuration. Typical use includes the ability to re-IPL the adjunct when necessary (ADJUNCT CP IPL).

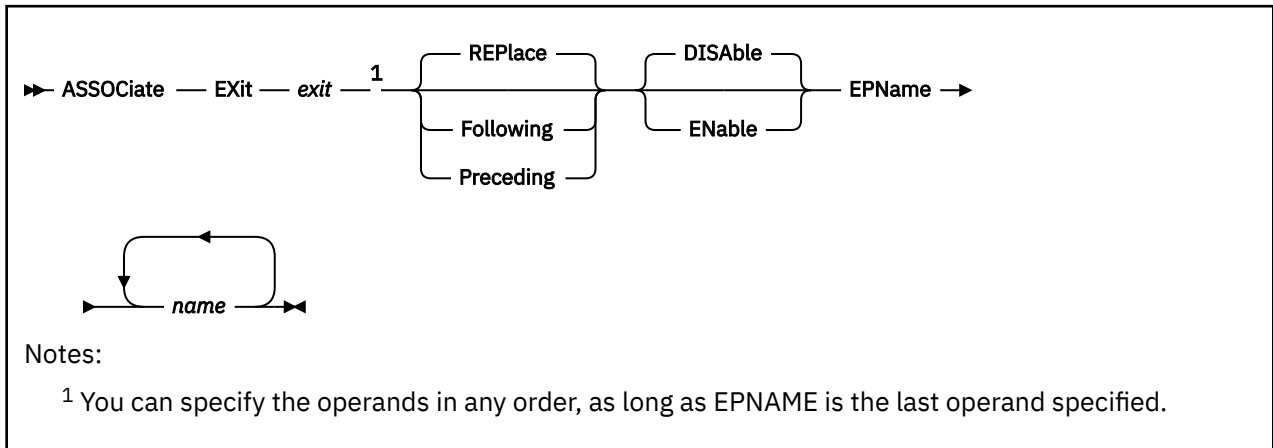
## Usage Notes

1. In order for you to create an adjunct, the feature must be enabled for the z/VM system and authorized for you by the system administrator. The administrator provides the virtual machine definition that specifies how the adjunct configuration is instantiated. Information on how to enable and set up adjuncts appears in *z/VM: CP Planning and Administration*, in the description of the FEATURE ENABLE ADJUNCTS configuration statement and the ADJUNCT directory statement.
2. To direct a command from the adjunct configuration to the principal, use the CPU prefix command, for example, CPU \* DEFINE STORAGE 4G or CPU ALL DISPLAY PSWG.
3. Use the QUERY ADJUNCT command to determine the state of your adjunct configuration.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP643E Adjunct feature is not enabled on this system
- HCP1461E No adjunct configuration defined
- HCP5768E *userid* Adjunct does not exist
- HCP5769E *userid* Adjunct already exists
- HCP5770E *userid* Adjunct is not exploitable

## ASSOCIATE EXIT



### Authorization

Privilege Class: A

### Purpose

Use ASSOCIATE EXIT to assign one or more entry points or external symbols to an exit point.

You can also assign entry points and external symbols to an exit point during initialization using the ASSOCIATE EXIT configuration file statement. For more information, see [ASSOCIATE EXIT Statement in z/VM: CP Planning and Administration](#).

### Operands

#### *exit*

is the number of the exit point to which you want to assign an entry point or external symbol. The variable *exit* must be a hexadecimal number between X'0000' and X'FFFF'.

#### **REPlace**

tells CP to replace the current list of entry point names and external symbols that are already associated with this exit point with the list specified after the EPNAME operand.

**Note:** The order that you specify the entry points and external symbols is the order in which CP will call them.

#### **Following**

tells CP to add the specified entry point names or external symbols to the end of the list of existing entry point names and external symbols that are already associated with the specified exit point number.

**Note:** The order that you specify the entry points and external symbols is the order in which CP will call them.

#### **Preceding**

tells CP to add the specified entry point names or external symbols to the beginning of the list of existing entry point names and external symbols that are already associated with the specified exit point number.

**Note:** The order that you specify the entry points and external symbols is the order in which CP will call them.

**DISable**

tells CP not to call the entry points and external symbols associated with this exit point until you enable it. (For more information about enabling exit points, see Usage Note “4” on page 21.) If omitted, DISABLE is the default.

**ENable**

tells CP to immediately start calling the entry points and external symbols associated with this exit point.

**EPName *name***

is the name (or names) of the entry point or external symbol that CP calls when encountering this exit point number. Each *name* must be a 1- to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

**Note:** The order that you specify the entry points and external symbols is the order in which CP will call them.

**Usage Notes**

1. Before you use the ASSOCIATE EXIT command, you should issue the CPXLOAD command or configuration file statement to load the customer-written CP routines for the exit point into the system execution space. These CP routines should contain the entry point names and external symbols that you will specify on the ASSOCIATE EXIT command.

If you do not issue the CPXLOAD command or statement first or you issue the ASSOCIATE EXIT command with an unknown entry point or external symbol, CP just ignores any entry point name or external symbol that it does not recognize and continues normal processing. If the unknown entry point or external symbol is part of a list associated with an exit point, CP continues processing the other members of the list. CP does not ignore an exit point because it cannot find one entry point or external symbol in the list. CP only ignores an exit point if it cannot find all the entry points and external symbols in the list.

For more information about the CPXLOAD command, see “CPXLOAD” on page 111.

For more information about the [CPXLOAD Statement](#), see *z/VM: CP Planning and Administration*.

2. To display whether there are any unknown entry points or external symbols associated with an exit point, use the QUERY UNRESOLVED command (see “[QUERY UNRESOLVED](#)” on page 1250).
3. If you issue the ASSOCIATE EXIT command and do not specify the ENABLE operand, CP will redefine the exit point with the information from your ASSOCIATE EXIT command, but will not call any of the entry points or external symbols associated with that exit point until you later enable it.
4. There are 2 ways to enable an exit point:
  - Enter another ASSOCIATE EXIT command and specify the ENABLE operand, or
  - Use the ENABLE EXITS command (see “[ENABLE EXITS](#)” on page 465).

By default, exit points are disabled. Thus, in general, you should follow any ASSOCIATE EXIT commands or statements with ENABLE EXITS commands or statements.

5. CP calls the entry points and external symbols for an exit point in the order that you specify them on the ASSOCIATE EXIT command, unless an entry point overrides this action. Any entry point can tell CP to change the normal processing flow by skipping one or all subsequent entry points or external symbols.
6. To display status and usage statistics information about a specific exit point, use the QUERY EXITS command (see “[QUERY EXITS](#)” on page 907).

**Note:** While processing the ASSOCIATE EXIT command (or statement), CP creates a CP exit block for the specified exit point. For a static exit point, CP does not create CP exit control blocks until you associate one or more entry points or external symbols with that exit point. If you try to issue a QUERY EXITS command against such an exit point, CP issues message HCP2752E as the response to

your QUERY EXITS command. For a dynamic exit point, QUERY EXITS responds with the definition of the exit, even if there are no entry points associated with it.

7. To display the address of the CP exit block for a specific exit point, use the LOCATE XITBK command (see [“LOCATE XITBK”](#) on page 649). Again, if you have not associated one or more entry points or external symbols with the specified exit point, there is no CP exit block for CP to locate and display. Instead, CP issues error message HCP2752E.
8. To display the address of the CP indirect call locator block for a specific exit point, use the LOCATE ICLBK command (see [“LOCATE ICLBK”](#) on page 614).
9. To change the definition of an existing dynamic exit point, or to remove the exit point from the system, use the MODIFY EXIT command (see [“MODIFY EXIT”](#) on page 700).
10. To stop CP from calling the entry points and external symbols associated with one or more exit points after defining those exit points, use the DISABLE EXITS command (see [“DISABLE EXITS”](#) on page 319).
11. To remove the customer-written CP routines from the system execution space:
  - a. Use the DISASSOCIATE command (see [“DISASSOCIATE”](#) on page 324) to revoke all entry point and external symbol assignments made with the ASSOCIATE EXIT command or configuration file statement.
  - b. Use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117) to unload the customer-written CP routines.
12. Using the ASSOCIATE EXITS command or configuration file statement, you can add to the list of entry points and external symbols for an exit point, or you can replace the list. For example, if you associate entry points ZZZA1 and ZZZA2 with an exit point:

```
associate exit 1 epname zzza1 zzza2
Ready;
```

and then you associate additional entry points YYYYB1 and YYYYC2:

```
associate exit 1 preceding epname yyyb1 yyyc2
Ready;
```

the final list of entry points would be:

```
YYYYB1 YYYYC2 ZZZA1 ZZZA2
```

13. To remove one or more entry points or external symbols from the list associated with an exit point, issue the ASSOCIATE EXIT command with the REPLACE operand and specify only those entry points and external symbols that you want CP to call.
14. If the exit point is disabled when you issue the ASSOCIATE EXIT command, CP immediately updates the list of entry points and external symbols with your changes. If the exit point is enabled, CP checks to see if the specified exit point is currently processing an exit request. If the exit point is not processing any requests, CP immediately updates the list with your changes. If the exit point is processing a request, CP waits until the current request is completed and then updates the list with your changes before processing any new exit requests.
15. After creating a CP exit block, CP will not erase that CP exit block until another IPL. Disabling the exit point affects certain fields in the CP exit block, but does not erase it. Disassociating the entry point names and external symbols erases those fields in the CP exit block, but does not erase the CP exit block itself.
16. For more information about user-defined exit points, see [IBM-Defined CP Exit Points](#) in *z/VM: CP Exit Customization*.

## Responses

Response 1:

To associate entry point HCPSRC00 with exit number F and to replace any existing entry point associations, enter the following:

```
associate exit f epname hcpsrc00
Ready;
```

Response 2:

To add entry point HCPSRC04 at the end of the current list for exit point 9C, enter the following:

```
associate exit 9c following epname hcpsrc04
Ready;
```

Response 3:

Using the QUERY EXITS command and the ASSOCIATE EXIT command, you can see how you are changing the list of entry points:

```
query exit 5
Exit Status      Calls   Returns Seconds
0005 Enabled      1       1 0.000122
      EPNAME    Attempts Calls Seconds
      HCPSRC00  1       1 0.000116
Ready;
```

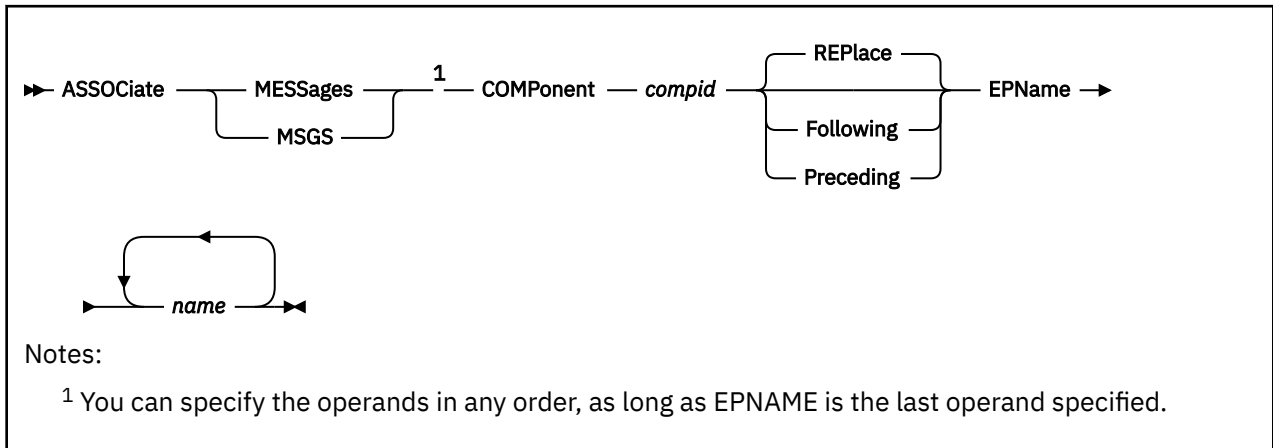
```
associate exit 5 following epname hcpsrc04
Ready;
```

```
query exit 5
Exit Status      Calls   Returns Seconds
0005 Enabled      1       1 0.000122
      EPNAME    Attempts Calls Seconds
      HCPSRC00  1       1 0.000116
      HCPSRC04  0       0 0.000000
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP013E Unknown entry point name *entry* cannot be associated with an exit point requiring resolution
- HCP376E Unknown entry point name *entry* cannot be associated with an exit point requiring resolution
- HCP2773I Entry point name *name* is unknown at this time; processing continues
- HCP6704E Missing token at end of line
- HCP6706E {Invalid entry point name - *name* | Invalid CP Exit number - *exit*}
- HCP6709E Too many items specified - *item*

## ASSOCIATE MESSAGES / MSGS



### Authorization

Privilege Class: A

### Purpose

Use ASSOCIATE MESSAGES or MSGS to assign an external symbol to a local message repository and to give CP information about how to select the messages in that repository.

You can also assign external symbols to a local message repository and give CP information about how to select messages in that repository during initialization using the ASSOCIATE MESSAGES or MSGS configuration file statements. For more information, see [ASSOCIATE MESSAGES / MSGS Statement in z/VM: CP Planning and Administration](#).

### Operands

#### COMPONENT *compid*

tells CP the component identifier to use when issuing one of the messages in the local message repository. The variable *compid* is a 1- to 3-character alphanumeric string. For example, the component ID for the system message repository (z/VM) is HCP, which is, by default, the last message repository in the search list. For more information, see Usage Note [“1” on page 25](#).

#### REPlace

tells CP to replace the current list of entry point names and external symbols that are already associated with this local message repository with the list specified after the EPNAME operand.

**Note:** The order that you specify the entry points and external symbols is the order in which CP will search them.

#### Following

tells CP to add the specified entry point names or external symbols to the end of the list of existing entry point names and external symbols that are already associated with the specified local message repository.

**Note:** The order that you specify the entry points and external symbols is the order in which CP will call them.

#### Preceding

tells CP to add the specified entry point names or external symbols to the beginning of the list of existing entry point names and external symbols that are already associated with the specified local message repository.



**Note:** The order that you specify the entry points and external symbols is the order in which CP will call them.

### EPName *name*

is the name (or names) of the entry point or external symbol that points to the data in the system execution space where the local message repository can be found. Each *name* must be a 1- to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

**Note:** The order that you specify the entry points and external symbols is the order in which CP will call them.

## Usage Notes

1. HCP is the standard component ID for z/VM messages. If you do not issue any ASSOCIATE MESSAGES or MSGS commands (or configuration file statements) for an entry point, HCPMES is, by default, the only message repository in the search list for that entry point.

If you do assign one or more local message repositories to an entry point, those repositories are added to the search list in the order that you specify (using the REPLACE, FOLLOWING, or PRECEDING operands) and, by default, HCPMES is the last message repository in the search list.

When the routines in that entry point issue a message, CP searches the first message repository in the search list. If CP finds the message in that repository, it issues the message and does not search any more repositories. If CP does not find the message, it continues searching through each repository until it finds the first occurrence of that message.

Using ASSOCIATE MESSAGES or MSGS commands (or configuration file statements), you can assign local message repositories which override existing z/VM messages in the HCPMES repository. Or, you can specify HCP as the component ID and move the z/VM message repository to a place in the search list other than last place.

2. Before CP can begin using your message repository, you must:
  - a. Generate your messages using the CMS GENMSG command. For more information about the [CMS GENMSG](#) command, see *z/VM: CMS Commands and Utilities Reference*.
  - b. Load the message repository file using the CPXLOAD command (see [“CPXLOAD”](#) on page 111).
3. If CP cannot find the message in the associated message repositories (or the entry point can't be found) when the message is being displayed, CP displays message substitution data, if any.
4. To display information about the local message repositories available on your system, use the QUERY CPLANGLIST command and specify the ASSOCIATED operand (see [“QUERY CPLANGLIST”](#) on page 826).
5. To remove the message repository file from the system execution space:
  - a. Use the DISASSOCIATE command (see [“DISASSOCIATE”](#) on page 324) to revoke the external symbol assignment made with the ASSOCIATE MESSAGES command or configuration file statement.
  - b. Use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117) to unload the repository.
6. To have CP use the message repositories associated with a specific component, you must specify that component ID on the COMPID keyword of the HCPCONSL macroinstruction. For more information about the HCPCONSL macroinstruction and about using your own message repositories, see [Creating, Controlling, and Using Local CP Message Repositories in z/VM: CP Exit Customization](#).
7. The ASSOCIATE MESSAGE configuration file statement has a DELAY/NODELAY operand so that you can delay associate processing until all CPXLOAD operations in the configuration file have been handled. For compatibility, you can specify these operands on the command line. However, CP ignores these operands, issues information message HCP2777I and continues processing the ASSOCIATE MESSAGE command immediately.

### Responses

Response 1:

To have CP assign an uppercase English message repository containing messages starting with OUR to entry point OURMESS1, enter the following:

```
associate msgs component our preceding epname ourmess1  
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP1003E *name* was not found.
- HCP1014E A required option is missing - *option*
- HCP2577E Message repository file *fn* is damaged
- HCP2777I {Operand|Option|Filemode} {DELAY|NODELAY|*filemode*} has been ignored
- HCP2778E Component identifier *compid* is not valid
- HCP6704E Missing token at end of line
- HCP6706E Invalid entry point name - *epname*

# AT



## Authorization

Privilege Class: A, B, C, D, E, F

There are restrictions on which CP commands can be issued with the AT command. For more information, see Usage Note “2” on page 27.

## Purpose

Use AT to issue a CP command on an active member system in an SSI cluster.

## Operands

### *sysname*

is the name of the SSI cluster member where the command is to be issued.

**\***

specifies that the command is to be issued on the SSI cluster member where the AT command is issued.

### **CMD *command\_string***

specifies the command string that will be issued on the designated member of the SSI cluster. Some commands have syntax requirements or restrictions when executed with the AT command. See the Usage Notes.

## Usage Notes

- CP commands that have an AT extension will work in an SSI cluster. However, when issued with the AT command, CP commands with an AT extension are rejected.
- Class restrictions for using the AT command include:
  - Class Any commands that are allowed: ASTERISK (\*), MESSAGE (user).
  - Class G commands that are allowed: QUERY CPLEVEL, QUERY CPOWNED, XAUTOLOG.
  - All non-class Any and non-class G commands are allowed except:
    - AT
    - AUTOLOG
    - DEFINE VFB-512
    - FLASHCOPY
    - FOR
    - GIVE
    - SAVESEG
    - SAVESYS
    - SEND
    - SET IPLPARMS
    - SET VTOD
    - SPXTAPE

- If an allowed command is authorized for multiple classes that include class Any or class G, the class Any or class G version cannot be issued with the AT command unless it is one of those specifically identified as allowed.
  - User-defined commands are processed according to the same privilege class rules.
3. Do not use an asterisk (\*) to represent a user ID in a command that is issued with the AT command. Specify the user ID. If you specify \*, it will be treated literally as a user ID, and the command will be rejected.
  4. In numerous CP commands, the command action is by default executed for the issuer of the command. When these commands are issued with the AT command to be executed on a different SSI member, it does not make sense to execute the command action for the issuer of the AT command. Because of this, some command options that are normally a default option must be specified when executed with the AT command. Commands that are affected by this have a syntax note explaining which default options must be specified when the command is issued with the AT command. Refer to the syntax for the command that you are issuing with the AT command for more details on any changes in default option handling when the command is executed with the AT command.
  5. The command text after the CMD keyword is, in most cases, passed in its entirety to the target system. An exception to this involves the LINEND setting of the CP TERMINAL command. For example, if the logical line-end symbol for the issuer's virtual machine is set to # (the default), the command at `tstmem2 cmd q cplevel # q cplevel` will display QUERY CPLEVEL results from the target system called TSTMEM2, followed by QUERY CPLEVEL results from the source system that the AT command was issued on. A logical line-end character cannot be passed as part of the command string to be executed on the target system. It will always be treated as part of the AT command.
  6. Some CP commands generate asynchronous responses. This means that the responses can be sent back to the issuer of the command after an indication has been sent back to the issuer that the command has completed initial processing.  
  
In general, if these commands are issued with the AT command, these asynchronous responses are not reflected back to the issuer of the AT command. This is consistent with how these responses are handled with commands issued with DIAGNOSE Code X'08' - Virtual Console Function. Any programs or processes that depend on these messages need to handle the fact that these asynchronous responses are not returned to the AT command issuer.
  7. The command issued with the AT command must be valid on the target system.

## Responses

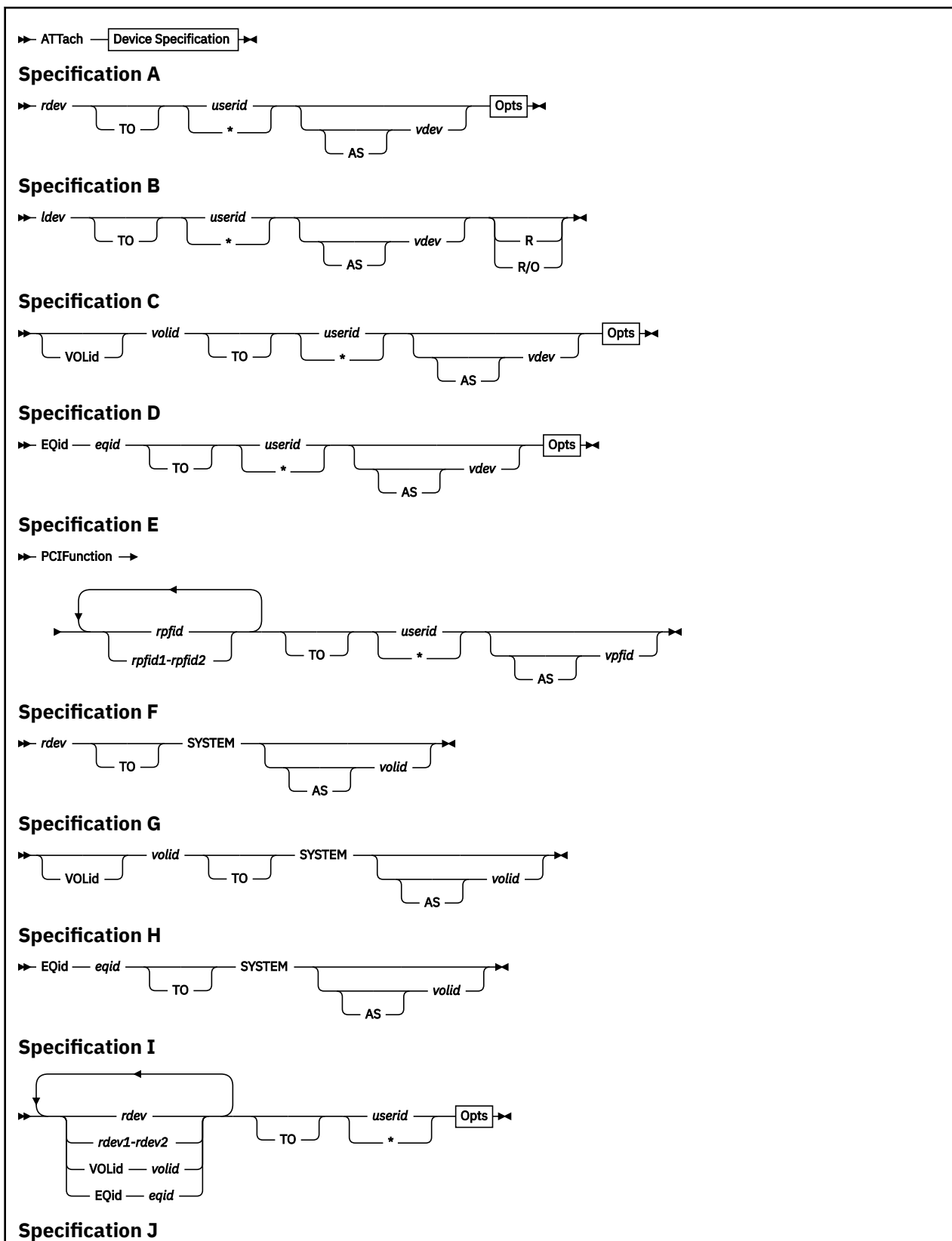
The responses received from the AT command are generated by the command that is executed on the SSI member system. Refer to the Responses section of the command that you are issuing with the AT command for response information.

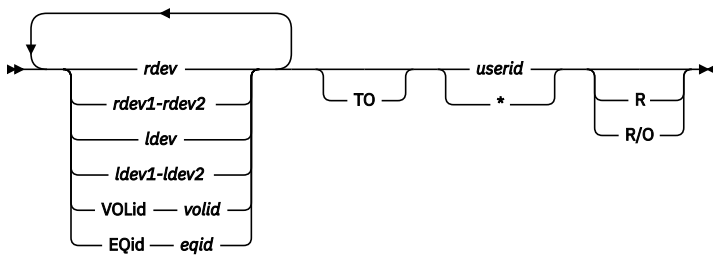
## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on.
- HCP1143E Member *sysname* is not in Joined status in this SSI cluster
- HCP1144E Unknown CP command executed with the AT command: *command\_string*
- HCP1145E Unknown command option specified with the AT command: *option*
- HCP1146E Unable to open connection for \*AT to *sysname*; rc=*rc*

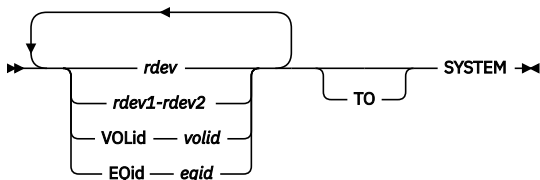
- HCP1149E Error communicating with another SSI member. It cannot be determined if AT executed remotely.
- HCP1927E System *sysname* is not a member of an SSI cluster
- HCP6525E The External Security Manager on *sysname* is unavailable.

# ATTACH

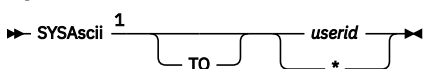




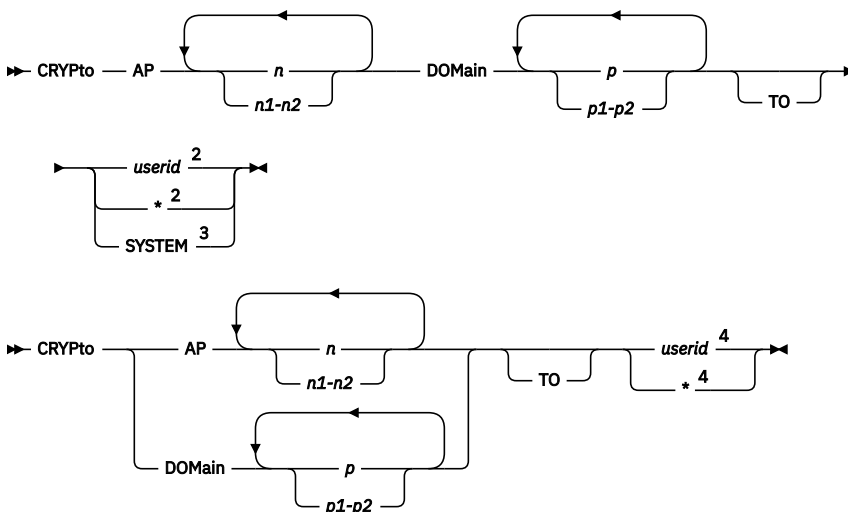
**Specification K**



**Specification L**



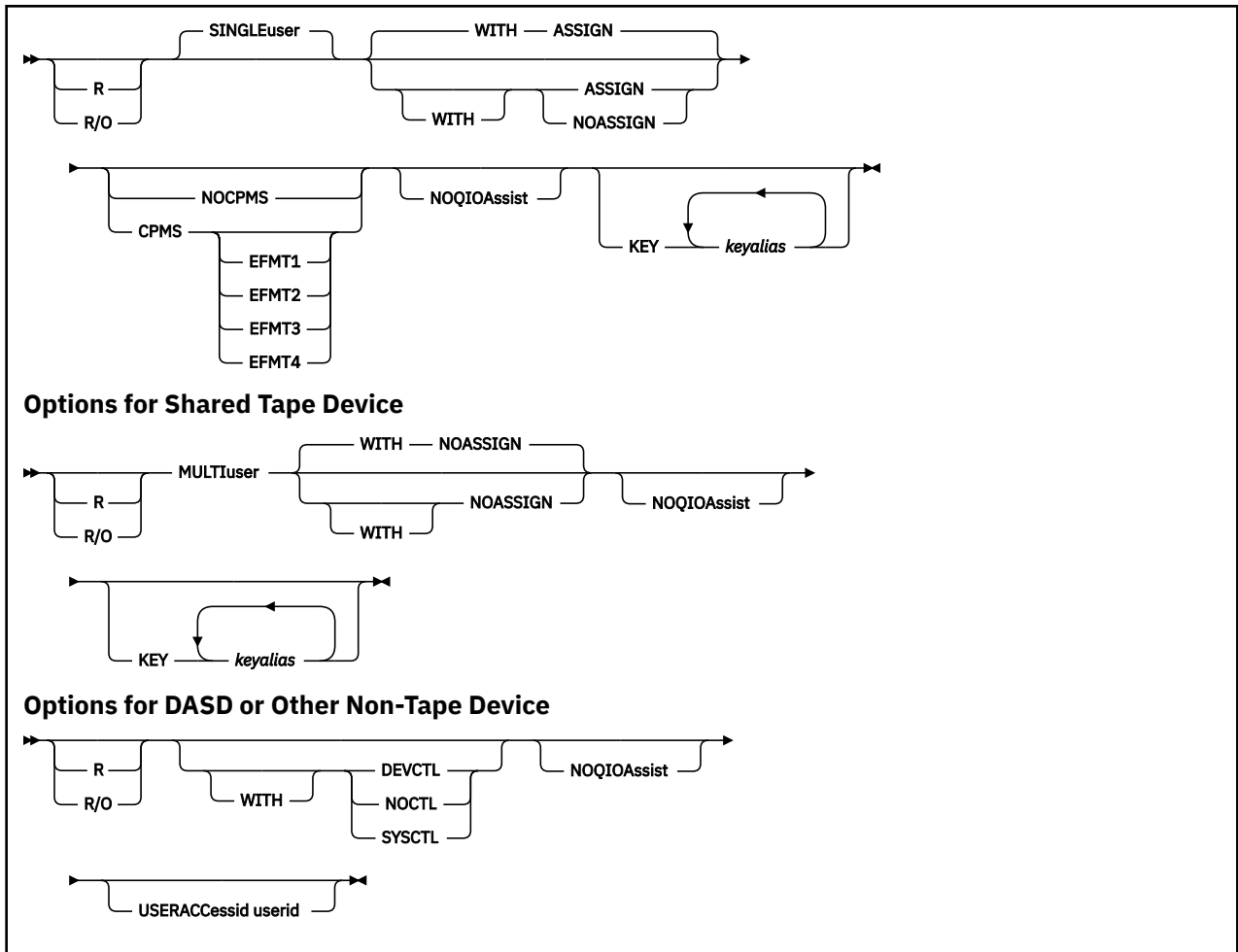
**Specification M**



**Notes:**

- <sup>1</sup> SYSASCII is for Linux® usage only.
- <sup>2</sup> The first ATTACH to *userid* (or ATTACH to \*) requires both crypto adapter (AP) and domain to be specified.
- <sup>3</sup> ATTACH TO SYSTEM always requires both adapter and domain to be specified.
- <sup>4</sup> After the initial set of dedicated crypto resources is attached to *userid* (or \*), you can specify only crypto adapters (AP) or domains, but not both.

**Options for Dedicated Tape Device**



## Authorization

Privilege Class: B

## Purpose

Use ATTACH to logically connect the following devices and resources:

- Connect a real or logical device to a virtual machine for exclusive use.
- Connect a real PCI function to a virtual machine for exclusive use.
- Connect a real tape device to a virtual machine for shared use.
- Connect DASD to the host system.
- Connect crypto resources to the system for shared use or to a virtual machine for dedicated use.

## Operands

**rdev**

**rdev1-rdev2**

specifies a real device number, a list of real device numbers, or a range of real device numbers to be attached to the specified user or to the CP system. Specify a real device number as 4 hexadecimal digits. You may omit leading zeros. The maximum number of devices allowed in a range is 256.

If you are not specifying a virtual device number or volume label for the real device that you are attaching, the virtual device number defaults to the real device number. You can combine device numbers, lists of device numbers, and ranges of device numbers on a single command line.



If you are specifying a virtual device number for the device, you can specify only one real device number when you enter the ATTACH command.

***ldev***

***ldev1-ldev2***

specifies a logical device number, a list of logical device numbers, or a range of logical device numbers to be attached to the specified user. Specify a logical device number as an "L" followed by up to 4 hexadecimal digits. You may omit leading zeros.

If you are not specifying a virtual device number for the logical device that you are attaching, the virtual device number defaults to the logical device number. You can combine device numbers, lists of device numbers, and ranges of device numbers on a single command line.

If you are specifying a virtual device number or volume label for the device, you can specify only one logical device number when you enter the ATTACH command.

**VOLid**

indicates that a volume serial number, which must follow this keyword, is used to identify the real device. If the volume serial number is valid as an operand of the command (e.g., is a valid device number), specify VOLID to indicate that the operand is a volume serial.

***valid***

specifies the 1- through 6-character volume serial number of a real DASD.

**EQid**

indicates that a device equivalency ID, which must follow this keyword, is used to identify the real device. If the EQID is valid and only one unattached device with this EQID exists, the device will be attached. If more than one unattached device exists with the specified EQID, the first unattached device found will be used.

***eqid***

specifies a string of 1 - 8 alphanumeric characters for a user-specified EQID or a string of 50 alphanumeric characters plus a dash ("-") for a system-generated EQID.

***userid***

**\***

is the virtual machine that is to receive the device or devices. That virtual machine must currently be logged on to the system. If you are attaching a device to your own virtual machine, you can specify your user ID as an asterisk (\*).

Use the keyword T0 when:

- You specify a user ID with three numbers or fewer in a range of addresses or multiple device addresses
- You specify a user ID that could also be a valid hexadecimal device address (for example, CE)
- You specify a user ID as "to" or "t".

***vdev***

is the virtual device number you want assigned to the device. The device number is assigned to an available virtual subchannel number, and may be as long as four digits.

**PCIFunction**

indicates that a PCI function ID, which must follow this keyword, is used to identify the real function.

***rpfid***

***rpfid1-rpfid2***

specifies a real PCI function ID or a range of PCI function IDs to be attached to the specified user. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. The total number of PCI functions specified cannot exceed 256.

**TO *userid***

is the virtual machine that is to receive the function or functions. That virtual machine must currently be logged onto the system. If you are attaching a function to your own virtual machine, you can specify your user ID as an asterisk (\*).

## ATTACH

### **AS *vpfid***

specifies the virtual PCI function ID you want assigned to the function. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. This option is only valid if a single *rpfid* is specified. If omitted, the virtual PCI function ID will be the same as the real PCI function ID specified.

### **SYSTEM**

makes the DASD available to CP for access to users' virtual disks, or for paging, spooling, and temporary disk functions. Indicates that crypto resources will be added to the system's shared pool.

### ***valid***

is the volume label of the volume on the device being attached. The system verifies the volume label on the device, so the label must be unique when compared with all other volumes attached to the system.

### **SYSAscii**

indicates that the real ASCII console is to be attached to the specified user.

### **SYSAscii TO *userid***

is the user ID of the virtual machine that is to receive access to the real ASCII console. The virtual machine must be currently logged on to the system. A user ID of SYSTEM is not valid.

### **SYSAscii TO \***

specifies that the virtual machine issuing the ATTACH command receives access to the real ASCII console.

### ***n***

#### ***n1-n2***

specifies a single crypto adapter number, a list of adapter numbers, a range of adapter numbers or a combination of these, to be attached. If you are specifying a list of numbers, insert at least one blank between the numbers you specify. If you are specifying a range of numbers, insert a hyphen (-) between the numbers you specify. The number is a decimal number between 0 and 255.

### ***p***

#### ***p1-p2***

specifies the crypto domain number, a list of domain numbers, a range of domain numbers or a combination of these, to be attached to the virtual machine or system. If you are specifying a list of domain numbers, insert at least one blank between the numbers you specify. If you are specifying a range of domain numbers, insert a hyphen (-) between the numbers you specify. The domain number is a decimal number between 0 and 255.

## **Options**

### **R/O**

can be used for DASD and tape devices that are attached to a user. This allows the system operator to apply read-only protection to a user's disk or tape.

CP ignores the R/O operand for any device number that does not correspond to a TAPE or DASD device.

### **SINGLEuser**

dedicates a real tape device to a single user. This is the default.

### **MULTIuser**

attaches a real tape device to be serially shared with other users. See Usage Note [“16” on page 37](#).

### **WITH ASSIGN**

specifies that the tape device you want attached to the user will be assigned to the user system. CP ignores this operand for any tape device that does not support ASSIGN and UNASSIGN CCWs.

### **WITH NOASSIGN**

specifies that the tape device you want attached to the user must not be assigned to the user system. This gives the guest the ability to control the assignment of the tape. CP ignores this operand for any tape device that does not support ASSIGN and UNASSIGN CCWs.

**Note:** When a tape attached with NOASSIGN is detached:

- The tape is *not rewound or unloaded*.
- If the tape was attached with SINGLEUSER (the default), CP does not change the ASSIGNED or UNASSIGNED status of the device. It is the responsibility of the guest to enter the ASSIGN CCW when it needs the tape and to enter the UNASSIGN CCW when it is finished using the tape.
- If the tape was attached with MULTIUSER, CP changes the status of the device to UNASSIGNED.

### **NOCPMS**

indicates that attach processing should not issue MODE SET CCWs to a tape device to initialize the Device Control Pages of the device. This option is ignored for any tape device that does not support the X'CB' MODE SET CCW. This option is not available on the DEDICATE User Directory Statement.

MODE SET CCWs are issued only if ASSIGN processing is also done. Thus, NOCPMS performs no function if NOASSIGN is specified.

The KEY option will issue MODE SET CCWs, but will respect the existing device settings if used in conjunction with the NOCPMS option.

### **CPMS**

indicates that CP will issue MODE SET CCW's with either default mode (no CPMS parameter) or with the specified mode parameter. Mode validation is done in respect of the tape hardware capabilities. CPMS and NOCPMS option cannot be issued together.

#### **EFMT1**

indicates the 512 track recording format.

#### **EFMT2**

indicates the 896 track recording format.

#### **EFMT3**

indicates the 1152 track recording format.

#### **EFMT4**

indicates the 2176 track recording format.

### **WITH DEVCTL**

means that the device accepts CCWs that have an effect on resources and functions directly related to the device. See Usage Note [“6” on page 36](#) for default values.

### **WITH NOCTL**

means that the device does not accept any CCWs that can exercise control of subsystem resources or functions, regardless of whether they directly relate to the device.

### **WITH SYSCTL**

means that the device accepts CCWs that have a direct global effect on subsystem resources and functions, not just on those related to the device. See Usage Note [“6” on page 36](#) for default values.

### **NOQIOAssist**

indicates that the devices are not eligible for Queued-I/O assist.

### **KEY keyalias**

prepares CP to enable a device's hardware encryption on behalf of a guest that is not able to manage the support itself. If a *keyalias* is specified (up to two are allowed), CP will use the encoding mechanism and key label described by the alias while setting up the encryption environment. If no aliases are supplied, the external key manager's default keys will be used.

### **USERACcesid userid**

to allow a user (guest) the ability to give FCP LUN access to the specified userid. This is a shortcut to redefining entries in the Access Control Table (ACT), and assumes the user has authority to the LUN as defined in the ACT. Please refer to Linux documentation for details on creating and maintaining the ACT for FCP configuration.

## Usage Notes

1. If you are attaching a DASD with logical addresses, the real device number you specify must be the base address for the device, and the virtual device number you specify must follow the addressing rules for base addresses.
2. The ATTACH command will not affect the various draining status indicators, so you can set the desired draining status before you attach a DASD to your system.
3. If you mount and ready a device that contains virtual minidisks after you have loaded VM onto the system, you must attach that device to the system if you want to use it. If you log on after the device has been attached, you gain access to the disk as provided by your entry in the system directory. If you were logged on before the volume was attached to the system, you must use the CP LINK command to gain access to the virtual disks.
4. CP processes all the device numbers you specify, even if it encounters errors when attempting to attach the devices. So, if you enter: `attach 108-10f user1 r` and real devices 10A and 10D were not ready and online, the following devices would be successfully attached: 108, 109, 10B, 10C, 10E, 10F. Messages tell you whether devices are successfully attached.
5. You can use the system directory DEDICATE statement to attach real devices to each virtual machine. Then, when users successfully log on, the devices (if online and available) are automatically attached to their virtual machine. This function saves Class B users the trouble of issuing multiple ATTACH commands for the devices.
6. Levels of control (SYSCTL/DEVCTL/NOCTL)
  - a. The default control level value is normally DEVCTL.
  - b. The SYSCTL, DEVCTL, and NOCTL operands apply only to DASD on a cached control unit. If one of these operands is specified for a noncached DASD, the device is attached and a response is issued stating that the operand was ignored.
  - c. Specifying a level of control (NOCTL, DEVCTL, or SYSCTL) for a DASD attached to a cache storage control unit authorizes CP to accept particular control CCWs.

Table 5 on page 36 shows the additional control CCWs CP accepts for a level of control. The DEVCTL column shows the CCWs CP accepts in addition to those it accepts for the NOCTL level of control. The SYSCTL column shows the CCWs CP accepts in addition to those it accepts for the DEVCTL and NOCTL levels of control.

Additional Control CCWs Accepted for DEVCTL	Additional Control CCWs Accepted for SYSCTL
<ul style="list-style-type: none"> <li>• Set subsystem mode               <ul style="list-style-type: none"> <li>– Activate caching for device</li> <li>– Deactivate caching for device</li> <li>– Activate DASD fast write</li> <li>– Deactivate DASD fast write</li> <li>– Force deactivate DASD fast write</li> </ul> </li> <li>• Perform Subsystem Function               <ul style="list-style-type: none"> <li>– Establish duplex pair</li> <li>– Terminate duplex pair</li> <li>– Suspend duplex pair</li> <li>– Direct I/O to one device of the duplex pair</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Set subsystem mode               <ul style="list-style-type: none"> <li>– Make cache available</li> <li>– Make cache unavailable</li> <li>– Force cache unavailable</li> <li>– Make NVS available</li> <li>– Make NVS unavailable</li> <li>– Activate cache fast write</li> <li>– Deactivate cache fast write</li> </ul> </li> <li>• Perform Subsystem Function               <ul style="list-style-type: none"> <li>– Destage Modified Tracks</li> <li>– Set Cache Allocation Parameters</li> <li>– Suspend/Resume Function</li> </ul> </li> </ul>

7. When an authorized user issues the GIVE command to transfer control of a tape drive, the ATTACH command is invoked.
8. The ATTACH command may be used to attach duplex volumes to a guest virtual machine or the system. Only the primary duplex volume may be attached to the system. Either the primary volume, the secondary volume, or both volumes may be attached to a guest virtual machine.  
When a duplexed volume is attached to a virtual machine (or to the system), the responses generated will indicate that the volume was either the primary duplex volume or the secondary duplex volume of a duplex pair.
9. For a guest virtual machine to establish a duplex pair, both of the affected volumes must be physically attached to the same 3990 storage control subsystem, and each volume must be logically attached to that guest with either DEVCTL or SYSCTL. In order for a guest virtual machine to suspend or stop a duplex pair, the guest must have DEVCTL or SYSCTL over the primary volume of the duplex pair.  
Duplexing is only allowed if both volumes are dedicated, or both are DEVNO-defined full-pack minidisks. CP does not allow a guest virtual machine to duplex a full-pack minidisk with a dedicated device.
10. The ATTACH command cannot be used to attach DEVNO-defined minidisks or secondary volumes of duplex pairs to the system.
11. The ATTACH command cannot be used to attach virtual disks in storage.
12. The ATTACH command will verify that the correct allocation map is used by the volume. Only FBA and devices with greater than 4K cylinders will use the extent-based allocation map. All other DASD will use the cylinder-based allocation map.
13. If a device is being throttled, you cannot attach it to the system as a CP-owned device. To display whether a device is being throttled, use the QUERY THROTTLE command (see [“QUERY THROTTLE” on page 1217](#)). To turn throttling off for a device, use the SET THROTTLE command (see [“SET THROTTLE” on page 1734](#)).
14. When attaching Parallel Access Volume DASDs:
  - To a guest, the base Parallel Access Volume must be attached first, before any associated alias Parallel Access Volumes. The associated alias Parallel Access Volumes can only be attached to the same guest as the base Parallel Access Volume.
  - To the system, the base Parallel Access Volume must be attached first, before any associated alias Parallel Access Volumes. Alias Parallel Access Volumes can be attached to the system, but are only exploited for VM I/O if they contain minidisk (PERM) or temporary disk (TDSK) allocations. Other CP volume allocations receive no benefit from system-attached alias Parallel Access Volumes.
15. HyperPAV aliases that are attached to the system are exploited by:
  - VM I/O, for minidisk (PERM) and temporary disk (TDSK) allocations
  - The CP paging subsystem, for directory (DRCT), page (PAGE), and spool (SPOL) allocations, when the feature has been enabled through one of the following:
    - In the system configuration file:
      - The PAGING\_ALIAS FEATURES statement (the preferred method)
      - The FEATURES ENABLE PAGING ALIAS statement
    - The SET PAGING command.
16. The MULTIUSER function is intended for guest operating systems that manage their own assignment of tape devices. It is not intended for CMS unless some external means of managing assignments or serializing access to the tape device among the sharing users is explicitly implemented. Third party assignment and multiple system assignment (Control Access CCW) are not supported.  
To share a tape device, a user must specify the MULTIUSER option when attaching the device, or the CP directory entry for that user must contain a DEDICATE statement for the device that includes the MULTIUSER operand. If the first user to log on attaches the device without specifying MULTIUSER, or has a DEDICATE statement for the device that does not include MULTIUSER, the device becomes

dedicated to that user. The device then cannot be attached as MULTIUSER by any user until it is detached by the user to whom it is dedicated.

The MULTIUSER function is valid only for 3480, 3490, and 3590 tape devices.

17. The USERACcesSid option applies only to first-level and second-level users. If it is attempted on a third-level or higher virtual machine, the request is translated into corresponding first-level and second-level information only.
18. The KEY option (and any supplied keyaliases) are valid only for 3592 tape drives that support hardware encryption. It is to be used only for those guests that do not exploit hardware encryption itself. If used with guests that do exploit hardware encryption, problems may occur such as missing interrupts or mode sense errors.
19. Encryption settings can persist on a device, via an ATTACH with encryption settings followed by a DETACH LEAVE or through an ATTACH MULTIUSER with encryption settings, as two examples. An ATTACH without encryption settings for either case will then pick up those settings on the device, while an ATTACH with encryption settings will fail if the encryption settings differ from what is currently associated with the device.
20. During an ATTACH TO SYSTEM command, a test is performed to determine if any non-PERM space is the Target of a FlashCopy relationship. If so, the command fails with message HCP2463E.
21. ATTACH fails if the device being attached has CHPID Virtualization on and the device is an unsupported DPS=NO DASD or TAPE. The following message is issued:
 

HCP6863E Device *ldev* has not been attached because it is a DPS=NO device and CHPIDVirtualization option is not specified for the user in the directory.
22. When issuing an ATTACH *rdev* to SYSTEM command for a CP owned volume, note that for SPOOL and DIRECTORY volume types extents are limited to cylinders 0-65519, and for PAGE volume types extents are limited to cylinders 0-1182005. TEMP and PARM disks can be located anywhere on the volume, but the PARM disk size is limited to 65520 cylinders. PERM space allocation is allowed anywhere on the volume.
23. When a CP-owned DASD volume with SPOOL extents is attached to a member of an SSI cluster, the volume will be automatically attached to the other active members where the device is online and not already attached. Also, if the volume is not in the CP-owned list on another active member where the device is being attached, the attach process will automatically add the volume to the CP-owned list on that member.
24. When ENFORCE\_BY\_VOLID has been enabled in the system configuration file, ATTACH commands will be rejected and message HCP6419E will be issued unless the ATTACH command specifies targets by VOLID.
25. The z/VM CP definition of a virtual function is a guest's view of a real function. This should not be confused with a virtual function associated with a PCI function where an adapter, identified by its PCHID, supports more than one PCI function.
26. The maximum value of the virtual function id number may be model dependent and limited by the machine and may also be limited by the guest operating system or application using the function. For example, z/OS 2.1 Communications Server IP Configuration allows a 2-byte hexadecimal value in the range 0-00FF.
27. When UID uniqueness enforcement is enabled by the SET IO\_OPT UID command, the attach of a PCI function will be rejected if another function with an identical UID value is already attached to the target guest.
28. The DEFINE CRYPTO APVIRTUAL command will give a virtual machine access to the system's shared crypto resources for the current logon session. The CRYPTO APVIRTUAL user directory statement will give a virtual machine access to the system's shared crypto resources at each logon.
29. The ATTACH CRYPTO TO SYSTEM command can be used to assign crypto resources to the system for shared use. This assignment can also be made during system IPL using a CRYPTO APVIRTUAL statement in the system configuration file.
30. The ATTACH CRYPTO TO userid command may be used to provide a virtual machine with dedicated access to one or more crypto resources for the duration of a logon session. This is the same authority

given to a virtual machine with a CRYPTO APDEDICATED statement in the directory entry. The CRYPTO APDEDICATED directory statement will give the virtual machine access to the specified crypto resources at each logon. The ATTACH CRYPTO command will give the virtual machine access to the specified crypto resources for the current logon session only.

31. Dedicated crypto resources must be assigned to a virtual machine as a rectangular resource set so that the resources available to a virtual machine are a cross-product of all assigned adapters and domains.

In order for a virtual machine to have dedicated access to a crypto resource, each specified domain number must be available for dedicated use by this virtual machine on all specified adapter numbers. If one or more domains on an adapter are not available for dedicated use by this virtual machine, then none of the domains on the adapter will be assigned. Any requested domains that are not assigned to the LPAR that the z/VM system is running on will be eliminated from the request and will not be reason to deny access to the remaining requested adapters or domains.

For example, assume that domain 1 on adapters 1, 2, and 3 are resources that are dedicated to virtual machine vmach1. You can then attach domain 2 to vmach1 only if domain 2 is available to vmach1 on all of adapters 1, 2, and 3. If domain 2 is not available on one or more of adapters 1, 2, or 3, then you cannot attach domain 2 to vmach1 on any of adapters 1, 2, or 3.

For more examples, see [“Examples” on page 39](#).

32. A crypto adapter must be configured in CCA coprocessor mode or accelerator mode in order to be attached to the system for shared use.
33. For information on planning and managing crypto resources on a z/VM system, see [Crypto Planning and Management in z/VM: CP Planning and Administration](#).

## Examples

Example 1:

```
ATTACH CRYPTO AP 2 3 DOMAIN 1 TO USER1
```

USER1 has:

Dom \ AP	000	001	002	003	004	005	006
000							
001			USER1	USER1			
002							
003							
004							
005							
006							
007							
008							

```
ATTACH CRYPTO AP 5 6 TO USER1
```

USER1 now has:

Dom \ AP	000	001	002	003	004	005	006
000							

*Table 7. Example 1B - Crypto Configuration (continued)*

Dom \ AP	000	001	002	003	004	005	006
001			USER1	USER1		USER1	USER1
002							
003							
004							
005							
006							
007							
008							

ATTACH CRYPTO DOMAIN 7 8 TO USER1

USER1 now has:

*Table 8. Example 1C - Crypto Configuration*

Dom \ AP	000	001	002	003	004	005	006
000							
001			USER1	USER1		USER1	USER1
002							
003							
004							
005							
006							
007			USER1	USER1		USER1	USER1
008			USER1	USER1		USER1	USER1

Example 2:

USER1 has:

AP 2 DOMAIN 1  
 AP 3 DOMAIN 1  
 and  
 AP 4 DOMAIN 2 is attached to the system.

USER1 has:

*Table 9. Example 2A - Crypto Configuration*

Dom \ AP	000	001	002	003	004
000					
001			USER1	USER1	
002					SHARED
003					



Table 9. Example 2A - Crypto Configuration (continued)

Dom \ AP	000	001	002	003	004
004					

ATTACH CRYPTO AP 4 to USER1

USER1 has:

Table 10. Example 2B - Crypto Configuration

Dom \ AP	000	001	002	003	004
000					
001			USER1	USER1	USER1
002					SHARED
003					
004					

ATTACH CRYPTO DOMAIN 2 TO USER1

This creates a conflict on adapter 4, domain 2 which is a shared crypto resource. Because domain 2 is not available on every adapter that USER1 currently has access to, the ATTACH command will fail.

## Responses

The receiver of the device attached, the user who issued the ATTACH command, and the primary system operator all receive messages indicating that the ATTACH was successful.

**Note:** The variable *type* is omitted from ATTACH responses when a range of device numbers or multiple device numbers are being reflected in the response. Control levels (for example, WITH xxxCTL) and PAV information (BASE and ALIAS) apply only to DASD; R/O applies to DASD and tapes.

In ATTACH responses, *type* is one of the following:

### Type

#### Meaning

#### DASD

Direct access storage device

#### FCP

SCSI Device Adapter

#### TAPE

Magnetic tape

#### RDR

Card reader

#### PRT

Printer

#### PUN

Card punch

#### GRAF

Display device

#### CTCA

Channel-to-channel adapter

## ATTACH

### OSA

Open Systems Adapter

### CTLR

Communications Controller

### MSC

Mass Storage Controller

### SWCH

Dynamic Switching Device

### DEV

Any other device

### LINE

Communication line

### SYSASCII

Integrated ASCII console

Response 1:

One of the following responses go to the receiver of the device:

```
{rdev... } ATTACHED TO userid
{rdev-rdev}
{ldev... }
{ldev-ldev}
```

– or –

```
{type rdev} ATTACHED TO userid vdev [R/O] [WITH SYSCTL] [PRIMARY DUPLEX ]
{type ldev} [WITH DEVCTL] [SECONDARY DUPLEX]
[WITH NOCTL ] [PAV BASE ]
[PAV ALIAS ]
[HYPERPAV BASE ]
[HYPERPAV ALIAS ]
```

informs the user that the devices have been attached to the user's virtual machine.

Response 2:

Depending on the command format the Class B user enters, this response indicates the devices attached to a specified user's virtual machine (and, if appropriate, the virtual device number, access mode, the level of control, and the duplex state, the PAV state, or the HyperPAV state for the device) or indicates that the device has been attached to the CP system.

One of the following responses go to the originator of the ATTACH command:

```
{rdev... } ATTACHED TO userid
{rdev-rdev}
{ldev... }
{ldev-ldev}
```

– or –

```
{type rdev} ATTACHED TO userid vdev [R/O] [WITH SYSCTL] [PRIMARY DUPLEX ]
{type ldev} [WITH DEVCTL] [SECONDARY DUPLEX]
[WITH NOCTL ] [PAV BASE ]
[PAV ALIAS ]
[HYPERPAV BASE ]
[HYPERPAV ALIAS ]
```

– or –

```
DASD rdev ATTACHED TO SYSTEM volid [PRIMARY DUPLEX ]
[PAV BASE ]
[PAV ALIAS ]
[HYPERPAV BASE ]
[HYPERPAV ALIAS ]
```

Response 3:

One of the following responses is sent to the primary system operator, if this operator did not issue the command. The response indicates that the device has been attached to either the specified user's machine (at virtual device number *vdev*) or to the CP system. The *originator* field designates the user ID of the user who issued the command.

One of the following responses go to the system operator:

```
{rdev... } ATTACHED TO userid BY originator
{rdev-rdev}
{ldev... }
{ldev-ldev}
```

– or –

```
{type rdev} ATTACHED TO userid vdev BY...con't
{type ldev}
```

Continued:

```
originator [R/O] [WITH SYSCTL] [PRIMARY DUPLEX ]
                  [WITH DEVCTL] [SECONDARY DUPLEX]
                  [WITH NOCTL ] [PAV BASE       ]
                              [PAV ALIAS      ]
                              [HYPERPAV BASE  ]
                              [HYPERPAV ALIAS ]
```

– or –

```
DASD rdev ATTACHED TO SYSTEM volid
BY originator [PRIMARY DUPLEX ]
                [PAV BASE       ]
                [PAV ALIAS      ]
                [HYPERPAV BASE  ]
                [HYPERPAV ALIAS ]
```

Response 4:

```
ctllevel IGNORED - INVALID FOR rdev
```

informs the originator of the command that control options have been specified for a noncached DASD.

***ctllevel***

is the level of control: SYSCTL, DEVCTL, or NOCTL.

***rdev***

is the real device number.

Response 5:

```
The directory on volume volume at address
rdev has been brought online.
```

indicates that a CP directory has been brought online during system initialization or during the process of attaching DASD to the system. The specified directory is the active system directory.

Response 6:

The following response goes to the user to whom the ASCII console is attached:

```
SYSASCII attached by originator
```

The variable *originator* designates the user ID of the person who entered the command.

The following response goes to the originator of the ATTACH command:

```
SYSASCII attached to userid
```

## ATTACH

If the primary system operator did not enter the command, the following response is sent to the primary system operator:

```
SYSASCII attached to userid by originator
```

The response indicates that the ASCII console has been attached to the specified user's virtual machine. The variable *originator* designates the user ID of the person who entered the command.

The receiver of the PCI function attached, the user who issued the ATTACH command, and the primary system operator all receive messages indicating that the ATTACH was successful.

Response 7:

The following response goes to the originator and receiver of the PCI function:

```
PCI FUNCTION rpfid ATTACHED TO userid upfid
```

Response 8:

The following response goes to the system operator:

```
PCI FUNCTION rpfid ATTACHED TO userid upfid BY originator
```

Response 9:

This response is sent to the command issuer and to the target when the target is not the command issuer or the primary system operator.

```
Crypto AP nnn Domain ppp attached to target
```

Response 10:

This response is sent to the primary system operator when the primary system operator is not command issuer.

```
Crypto AP nnn Domain ppp attached to target by originator
```

Response 11:

This response is sent to the originator of the command.

```
Crypto AP nnn Domain ppp attached to SYSTEM
```

Response 12:

This response is sent to the primary system operator when the primary system operator is not the originator of the command.

```
Crypto AP nnn Domain ppp attached to SYSTEM by originator
```

### ***nnn***

is the adapter number.

### ***ppp***

is the domain number.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option| command contains extra option(s) starting with option}*
- HCP009E Invalid range - *range*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid

- HCP021E A real PCI function ID was not supplied or it is invalid.
- HCP022E A virtual PCI function ID was not supplied or it was invalid.
- HCP023E VOLID missing or invalid
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP040E PCI function *rpfid* does not exist
- HCP045E *userid* not logged on
- HCP046E *type rdev | rpfid* offline
- HCP046E SYSASCII offline
- HCP048E No device with EQID *eqid* exists.
- HCP049E *type vdev* in use
- HCP120E *type {rdev | ldev | rpfid}* not {attached | given}; *userid* { *vdev* | *vpfid* already defined
- HCP122E *type {rdev | ldev | rpfid}* already attached to *userid*
- HCP122E SYSASCII already attached to *userid*
- HCP125E DASD *rdev* EQID *eqid* already attached to system
- HCP125E DASD *rdev* volid *valid* already attached to system
- HCP126E DASD *rdev* error reading volid
- HCP127E DASD *rdev* volid *valid* does not match
- HCP128E DASD *rdev* error reading allocation record
- HCP130E No devices with EQID *eqid* are available to attach with the parameters provided.
- HCP142E *type rdev* not drained
- HCP153E Device {*vdev|range*} exceeds maximum number of virtual devices
- HCP155E Device *rdev* has not been attached to the system because it is not a DASD device.
- HCP263E Too many operands were supplied for this command.
- HCP845E DASD *rdev* not attached; paging addresses offline
- HCP1014E A required option is missing
- HCP1100E A logical device number was not supplied or it is invalid.
- HCP1101E DASD *rdev* is not supported as a system-owned device.
- HCP1102E DASD *rdev* has not been attached because the paging subsystem is already owned by *userid*.
- HCP1103E DASD *rdev* has not been attached because a real DASD paging subsystem must map to only one virtual paging subsystem.
- HCP1104E DASD *rdev* has not been attached because multiple real DASD paging subsystems must not map to the same virtual DASD paging subsystem.
- HCP1105E Tape *rdev* not attached; tape assigned elsewhere.
- HCP1107E Attach failed; I/O error on tape *rdev*
- HCP1107E Attach failed; encryption requires tape *rdev* to be positioned at the beginning-of-tape
- HCP1113I DASD *rdev* was not ATTACHED to the SYSTEM because this operation would conflict with an MDISK definition for user *userid*.
- HCP1114I DASD *rdev* was not ATTACHED to the SYSTEM because it is the secondary in a duplex pair.
- HCP1116E Tape *rdev* not attached; the tape is being rewound.
- HCP1117I DASD *rdev* was not ATTACHED to the SYSTEM because it is the secondary in a PPRC pair or a cascading intermediate PPRC volume.
- HCP1124I An allocation map containing invalid data was found on DASD *rdev* volume *valid*; the device is not attached.

- HCP1125I The allocation map for DASD *rdev*, volume *valid* contains an area that is currently undefined. That area will be ignored.
- HCP1128E Device *rdev* not attached; MULTIUSER or SINGLEUSER not supported on this device.
- HCP1128E Device *rdev* not attached; hardware encryption not supported on this device.
- HCP1128E Device *rdev* not attached; a mismatch in hardware encryption settings was detected.
- HCP1128E Device *rdev* not attached; *user* is not authorized.
- HCP1128E Device *rdev* not attached; unsupported mode specified.
- HCP1129I *type rdev* has invalid volume label.
- HCP1129I *type rdev* has invalid volume prefix.
- HCP1150E DASD {*rdev|vdev*} is not a valid base exposure.
- HCP1168E PCI function *rpfid* cannot be attached to *userid* because a function with the same UID is already attached to that *guest*.
- HCP1280E Device *vdev* not usable; reserve lost [by *userid* as *vdev*]
- HCP1281I Path *channel* to device *rdev* now offline; path group ID mismatch.
- HCP1401I Expanded storage is not supported.
- HCP1561I Real device *rdev* forced R/O; unauthorized for R/W.
- HCP1728E AP Crypto resources are not installed on this system.
- HCP1728E Crypto AP *apnum* Domain *domnum* is not assigned to this system.
- HCP1728E Crypto AP *apnum* is not assigned to this system.
- HCP1728E Crypto Domain *domnum* is not assigned to this system.
- HCP1737E Crypto AP *apnum* Domain *domnum* cannot be attached to SYSTEM because it is attached to a user.
- HCP1737E Crypto Domain *domnum* cannot be attached to SYSTEM because it is not configured in coprocessor or accelerator mode.
- HCP1737E Crypto AP *apnum* Domain *domnum* cannot be attached to SYSTEM because it does not match the currently defined shared type.
- HCP1738E Only APs or Domains can be specified, not both, when attaching AP crypto resources to a user that already has dedicated AP crypto resources assigned.
- HCP1738E Both APs and Domains must be specified when attaching AP crypto resources to a user that does not have dedicated AP crypto resources already assigned.
- HCP1739E Dedicated AP crypto resources cannot be attached to an APVIRTUAL crypto user.
- HCP1747E Crypto AP *apnum* Domain *domnum* is already attached to SYSTEM.
- HCP1748E Crypto AP *apnum* cannot be attached because not all domains assigned to this user are free on this AP.
- HCP1748E Crypto Domain *domnum* cannot be attached because this domain is not free on all APs assigned to this user.
- HCP1748E Crypto AP *apnum* Domain *domnum* cannot be attached.
- HCP1749E AP crypto resources on AP *apnum* are already attached to this user.
- HCP1749E AP crypto resources on domain *domnum* are already attached to this user.
- HCP1753E There is no directory online because the directory on volume *valid* at address *rdev* is not valid.
- HCP1831E The command could not be completed because of a temporary condition. Try again later.
- HCP2462I *extenttype* extent on CP-OWNED volume *volser*, tracks *cccc/hh* to *cccc/hh*, is the Source for FlashCopy Relationship *sequence*.
- HCP2463E *extenttype* extent on CP-OWNED volume *volser*, tracks *cccc/hh* to *cccc/hh*, is the Target for FlashCopy Relationship *sequence*.

- HCP2768E Missing user ID
- HCP2768E Missing crypto domain number
- HCP2768E Missing crypto AP number
- HCP2777I Option *option* has been ignored.
- HCP2826E *operand* contains invalid UTF-8 data: *data*.
- HCP2890E CSE I/O error, code *cc*, volume *valid*; LINK commands may fail.
- HCP3173E Device *device* is reserved for system use.
- HCP5780E Adjunct does not support the *command subcommand* command
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6005E Option *option* is not supported by z/VM.
- HCP6201E Volume *volser* cannot be associated with a dump-only CP-owned list entry.
- HCP6202E The {*spool|temporary disk|directory*} cylinders were ignored on system volume *valid*.
- HCP6275E An error occurred writing to the subchannel for device *rdev*, code = *code*.
- HCP6419E DASD *rdev* has not been attached because the system requires it to be attached with the VOLID parameter.
- HCP6430E ATTACH cannot complete because the associated PAV base is quiesced for Hyperswap.
- HCP6430E ATTACH cannot complete because there is at least one base device in this HyperPAV pool that is quiesced for Hyperswap.
- HCP6643E Device *rdev* not attached to SYSTEM. Device contains spool extents and is already attached on system *system*.
- HCP6643E Device *rdev* not attached to SYSTEM. Device contains spool extents and is in slot *slot*. System *system* contains valid *valid*.
- HCP6643E Device *rdev* not attached to SYSTEM. Device contains spool extents and is in slot *slot*. This slot does not exist on system *system*.
- HCP6643E Device *rdev* not attached to SYSTEM. Device offline on system *system*.
- HCP6643E Device *rdev* not attached to SYSTEM. SSI cluster error, code *code*.
- HCP6644E Device with EQID *eqid* not attached to SYSTEM. Device contains spool extents and has been attached on other systems in the SSI cluster and could not be attached to this system.
- HCP6645E SSI cluster is in safe mode; command rejected.
- HCP6706E Invalid user ID
- HCP6706E Invalid Crypto Domain number
- HCP6706E Invalid Crypto AP number
- HCP6790I ATTACH processing halted. The next function that would have been processed is *rpfid*
- HCP6807E Device *rdev* is being throttled and cannot be attached to the system as a CP-owned device.
- HCP6860E Device *rdev* cannot be attached to the system because it is an alias Parallel Access Volume for which the base is attached to a user.
- HCP6860E Device *rdev* cannot be attached to the system because it is an alias Parallel Access Volume for which the base is free.
- HCP6863E Device *rdev* has not been attached because it is an alias Parallel Access Volume for which the base is free.
- HCP6863E Device *rdev* has not been attached because it is an alias Parallel Access Volume for which the base is in use by another user.
- HCP6863E Device *rdev* has not been attached because it is an alias Parallel Access Volume for which the base is in use by the system.
- HCP6863E *userid*: Device *rdev* has not been attached because it is a DPS=NO device and the CHPID Virtualization option is specified for the user in the directory.

## ATTACH

- HCP6863E Device *rdev* has not been attached because it is an EDEVICE alias.
- HCP9968E The key alias contains leading or trailing blanks or an empty string.



## ATTN

---

▶- ATTN -▶

### Authorization

Privilege Class: G

### Purpose

Use ATTN to interrupt the operation of your virtual machine.

### Usage Notes

1. The CP REQUEST command performs the same function as ATTN; you can use the two commands interchangeably.
2. The ATTN command performs an implied BEGIN.
3. CP ignores the ATTN and REQUEST commands if another attention interrupt is pending for your virtual machine, or if your virtual machine is not enabled for interrupts.

### Messages

- HCP002E Invalid operand - *operand*

## AUTOLOG

```

▶▶ AUTOLOG — userid — password 1
                                     └─ console_input_data ─┘
    
```

Notes:

<sup>1</sup> The *password* operand is not accepted if password suppression is active.

### Authorization

Privilege Class: A, B

### Purpose

Use AUTOLOG to log on another virtual machine automatically.

Before a virtual machine can be automatically logged on, it must be defined in the system directory, and you must know its password. After logon, it operates in disconnect mode.

**Note:** You can also use XAUTOLOG to log on another virtual machine.

XAUTOLOG has parameters for defining the virtual machine and changing command synchronization. Class A or Class B users do not need a password to use XAUTOLOG; Class G users must be authorized in the other virtual machine system directory.

Read the XAUTOLOG command description for more information about the additional functions available.

### Operands

#### *userid*

identifies the 1-character to 8-character name of the virtual machine you are logging on.

#### *password*

identifies the password of the virtual machine you are logging on.

To determine whether password suppression is active for this command, use the QUERY PASSWORD AUTOLOG command.

If password suppression is active, all data following *userid* is taken as *console\_input\_data*. Wait for the ENTER PASSWORD prompt before you type in the password. The password is automatically masked for system security.

If password suppression is not active, the first data item following *userid* is taken as the password, and any data following the password is taken as *console\_input\_data*.

#### *console\_input\_data*

identifies the data you want used as virtual console input in response to the first read issued by the user virtual machine. You may use the remainder of the command line to enter console data. The entire line, including any logical-line-end characters (#), is passed to the virtual machine.

**Note:** # refers to the logical-line-end character currently in effect for the virtual machine in which the AUTOLOG command is being used.

If you are using CMS and you specify *console\_input\_data* with the logical line-end character (#) on the command line, CMS truncates the AUTOLOG command at the first logical line-end character. To prevent this, enter a logical escape character before the logical line-end character, or enter #CP, where # is your logical line-end character, before the AUTOLOG command itself.

## Usage Notes

1. The virtual machine that you log on using AUTOLOG must have an IPL statement in its system directory unless it is a CF Service Machine (has OPTION CFVM in the directory).
2. A virtual machine is allowed to be logged on with AUTOLOG even though the number of users already logged on is equal to or greater than the maximum allowed by the MAXUSERS operand of the FEATURES system configuration statement or by the SET MAXUSERS command if the user directory for the virtual machine was created using the IGNMAXU operand of the OPTION directory statement.
3. The virtual machine you log on using AUTOLOG operates in disconnect mode. The same restraints that apply to any disconnected machine apply to virtual machines logged on with AUTOLOG.
4. The password suppression setting for the AUTOLOG command is controlled by the PASSWORDS\_ON\_CMDs operand of the FEATURES system configuration statement or by the SET PASSWORD command.
5. CP disables the AUTOLOG command for the rest of a user logon session if the user enters too many incorrect passwords while trying to logon another user ID automatically. The initial system limit depends on how the JOURNALING system configuration statement is coded. See the [JOURNALING Statement](#) in *z/VM: CP Planning and Administration* for more information. Use the QUERY CMDLIMIT command to see the system limit value set by your installation.
6. A secondary user ID may be specified on the user system directory CONSOLE statement. All console messages are routed to the secondary user ID, if it is functional, after the user is automatically logged on.
7. If you use AUTOLOG in an EXEC or module, you must put the password in the command or set the DIAGNOSE X'08' control bit to indicate that you want a prompt. See *z/VM: CP Programming Services* for details about [DIAGNOSE X'08'](#).
8. There is no password authorization for an AUTOLOG of a *userid* defined with the NOPASS operand, unless an Extended Security Manager (ESM) is installed. When an ESM is installed, password authorization may be required despite the NOPASS operand. Refer to documentation provided by your ESM for more information.
9. CP typically presents *console\_input\_data* to CMS through an interruption sequence before a CMS program has requested a console read. CMS places the *#console\_input\_data* into its terminal input queue in anticipation of some subsequent request to read from the console. This subsequent request is usually the PARSE EXTERNAL instruction in the SYSPROF EXEC, unless the SYSPROF EXEC is bypassed or the AUTOOCR parameter was used at IPL time. The EXTERNALS() VM REXX built-in function can be used to test for the existence of *console\_input\_data* before the PARSE EXTERNAL is issued.
10. Users defined with the LBYONLY operand in the password field of their USER or IDENTITY statement in the system directory may be restricted from performing functions that require password validation. When an External Security Manager (ESM) is not installed, users defined with the LBYONLY operand cannot be logged on by the AUTOLOG command. When an ESM is installed, refer to the documentation provided by your ESM for more information.

Refer to the [USER Directory Statement](#) or [IDENTITY Directory Statement](#) in *z/VM: CP Planning and Administration* for more details on the LBYONLY operand.

## Responses

Response 1:

```
ENTER PASSWORD
(IT WILL NOT APPEAR WHEN TYPED):
```

is the response when you must enter a password. Enter the password of the user ID you want to log on automatically.

Response 2:

## AUTOLOG

```
AUTO LOGON *** userid
USERS = nnnnn
```

is the response AUTOLOG sends to the issuer of the command when that person is not the system operator. This response is generated by successful command completion.

**\*\*\***

indicates no assigned line address for the virtual console.

***userid***

identifies the virtual machine initiated by AUTOLOG.

**USERS = *nnnnn***

identifies the number of logged-on users.

Response 3:

```
hh:mm:ss AUTO LOGON *** userid
USERS = nnnnn BY {SYSTEM }
                {userid2}
```

is the response AUTOLOG sends to the system operator when the command completes successfully.

**\*\*\***

indicates no assigned line address for the virtual console.

***userid***

identifies the virtual machine initiated by AUTOLOG.

**USERS = *nnnnn***

identifies the number of logged-on users.

**SYSTEM**

indicates that the AUTOLOG process has occurred as part of VM system initialization. Typically, your primary system operator's virtual machine is logged on as part of the system initialization process.

***userid2***

identifies the user issuing the AUTOLOG command. If *userid2* and *userid* are the same, then CP has autologged this virtual machine so that an APPC private server connection can be made.

**Response 4:**

```
hh:mm:ss USER DSC LOGOFF AS userid
USERS = nnnnn Forced
```

is the response to the system operator if an invalid IPL statement is processed for the user specified in the AUTOLOG command.

***userid***

identifies the virtual machine initiated by AUTOLOG.

**USERS = *nnnnn***

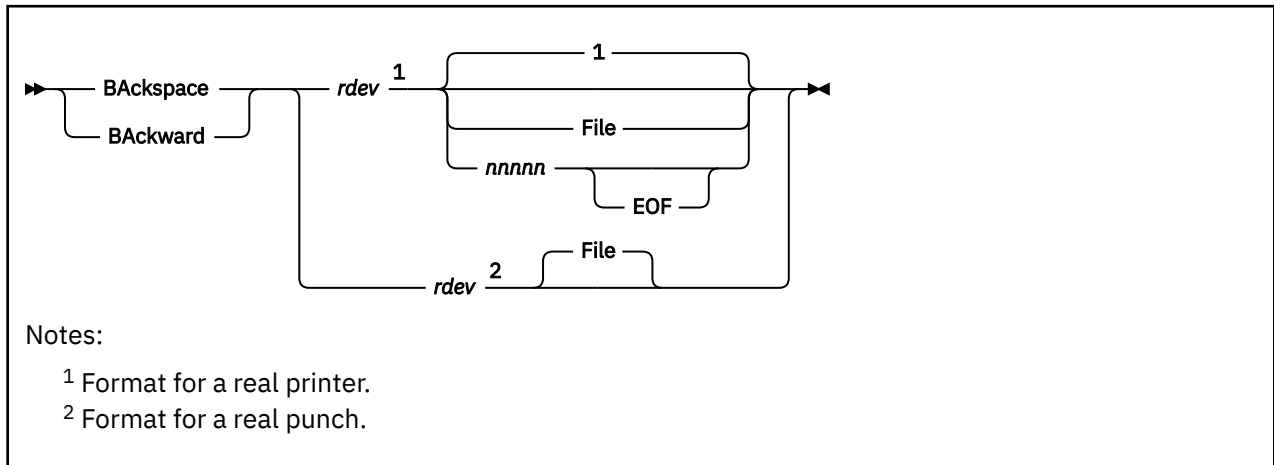
identifies the number of logged-on users.

## Messages

- HCP020E Userid missing or invalid
- HCP025E Storage missing or invalid
- HCP052E [XAUTOLOG failed for *userid*:] Error in CP directory
- HCP053E [XAUTOLOG failed for *userid*:] *userid* not in CP directory
- HCP054E [XAUTOLOG failed for *userid*:] Already logged on {*line vdev*|*GRAF vdev*|*LUNAME vdev*| *device vdev*|*LDEV vdev*|disconnected| *SYSTEM sysname*|SYSC}
- HCP059E {XAUTOLOG|AUTOLOG} failed for *userid* - {IPL missing|IPL failed|accounting errors|incorrect password}

- HCP093E Storage size requested *reqsize* exceeds maximum allowed on this processor *maxsize*. Size set to maximum allowed.
- HCP115E {LINK|AUTOLOG|XAUTOLOG} invalid; excessive incorrect passwords
- HCP145I User *userid* {at *rdev*} issued {a LINK|a LOGON|an AUTOLOG specifying *userid* *userid* |an APPCVM CONNECT specifying *userid* *tuserid*} with an invalid password *nnnn* times. The limit is *limit*.
- HCP1472I One or more directory COMMAND statements were found for the CFVM user and were discarded. Directory COMMAND statements are not allowed for CFVM users.
- HCP205I The virtual=real area is not available because it was not generated.
- HCP266E An account number was not supplied or is invalid.
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP780E [XAUTOLOG failed for *userid*:] - Maximum password attempts exceeded, try again later.
- HCP1502E [XAUTOLOG failed for *userid*] The directory entry for this *userid* is in use.
- HCP6050E Your *userid* is not authorized to automatically logon *userid* *userid*
- HCP6525E Authorization request failed.

## BACKSPACE/BACKWARD



### Authorization

Privilege Class: D

### Purpose

Use BACKSPACE to restart or reposition real device output (printer or punch).

### Operands

#### *rdev*

identifies the real device number of the punch or printer to be backspaced.

#### **File**

specifies that the output spool printer or punch file is to be restarted from the beginning. For punch output, FILE is the default.

#### *nnnnn*

specifies the number of pages (in decimal, to a maximum of 32,767) to be backspaced on the printer. If you do not specify a value, the printer backspaces to the beginning of the page being printed when the command is received. For printer output, the default value is 1.

#### **EOF**

specifies that backspacing is to start from the end of the spool file and then continue backward for the requested number of pages. Printing begins again from this new position.

### Usage Notes

1. BACKSPACE is identical to BACKWARD.
2. If you enter a BACKSPACE command while an intervention-required condition is pending on your printer or punch, the command does not take effect immediately. Once the intervention-required condition is cleared and the current outstanding I/O operation completes, the BACKSPACE command is executed.
3. In BACKSPACE operations, pages are counted by looking for page ejects. If a file does not contain page ejects, 66 output lines, including blank lines, are treated as a "page."
4. This command may re-print more than the specified number of pages. The number of pages re-printed depends on the data in the file and the position within the file of the print processing at the time the BACKSPACE/BACKWARD command was issued.

## Responses

Response 1:

```
{PRT} rdev BACKSPACE userid FILE spoolid RECS {nnnn} COPY {*nnn} SEQ sss CL c typ
{PUN}                                     {nnnK}   { nnn}
                                           {nnnM}
```

### PRT

### PUN

indicates device type.

### rdev

identifies the real device number.

### userid

identifies the user whose file was active and backspaced.

### FILE spoolid

identifies the system-assigned spool ID by which the system refers to the file.

### RECS nnnn

### RECS nnnK

### RECS nnnM

identifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines in thousands rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million.

### COPY \*nnn

### COPY nnn

identifies the remaining number of file copies to be produced. The asterisk (\*) is for the 3800 printer and indicates that the printer copies each page *nnn* times before going on to the next page.

### SEQ sss

identifies the sequence number for the file on the printer or punch.

### CL c

is the spool file class.

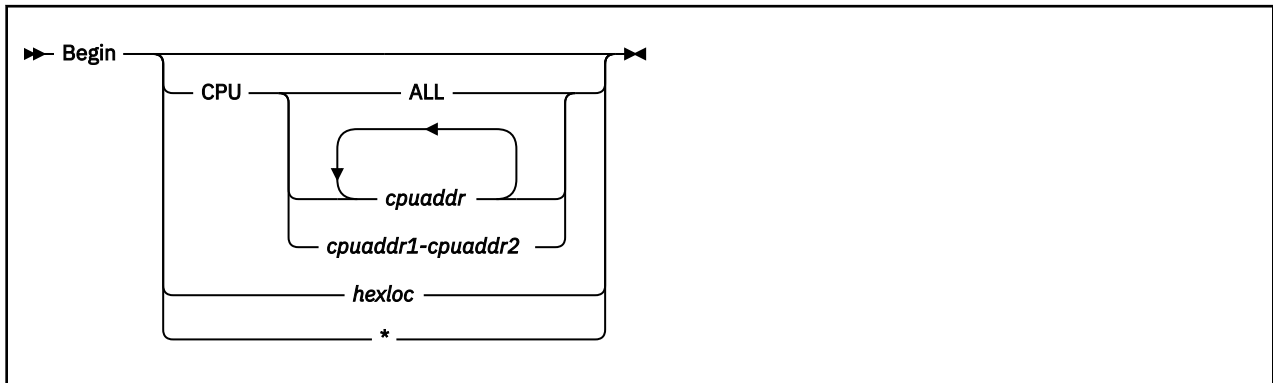
### typ

shows the device that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP, or SYS.

## Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*}
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP046E *type rdev* offline
- HCP140E *type {rdev/ldev}* attached to *userid*
- HCP141E *type rdev* not active
- HCP417E *{rdev/vdev}* is not an output spooling device.
- HCP418E *{rdev/vdev}* is not a spooled device.
- HCP9999E This is a dummy message.

## BEGIN



### Authorization

Privilege Class: G

### Purpose

Use BEGIN to resume command execution in a virtual configuration.

### Operands

#### CPU ALL

specifies all virtual processors currently defined.

#### CPU *cpuaddr*

terminates a hard stop for the specified processors without affecting any others.

#### CPU *cpuaddr1-cpuaddr2*

specifies a nonwrapping pair of addresses separated by a hyphen (-) and not separated by any blanks.

#### CPU *cpuaddr cpuaddr*

identifies a list of addresses delimited by blanks, a range of addresses delimited by blanks, or both.

#### *hexloc*

identifies the hexadecimal storage location where execution is to begin for the processor on which the command is issued. Command execution terminates a hard stop. To begin multiple processors at specified hexadecimal locations, use the CPU command with the BEGIN command.

If your virtual machine is in z/Architecture mode, up to 16 hexadecimal digits are valid. An underscore is permitted to separate the hex digits into two groups for easier reading if the location is larger than 8 digits, for example:

```
39A40B_AF8216CE
```

\*

terminates hard stop for the processor on which it is issued. No other virtual processors are affected.

### Usage Notes

1. If you enter BEGIN without operands, it terminates a soft stop on all virtual processors in your configuration. If all of your processors are in hard stop, BEGIN terminates hard stop for the processor on which it is executed.
2. If you enter BEGIN without a starting storage location, command execution begins at the location the current PSW points to. Unless the PSW was altered after you entered CP command mode, the location stored in the PSW is the location where the virtual machine stopped.

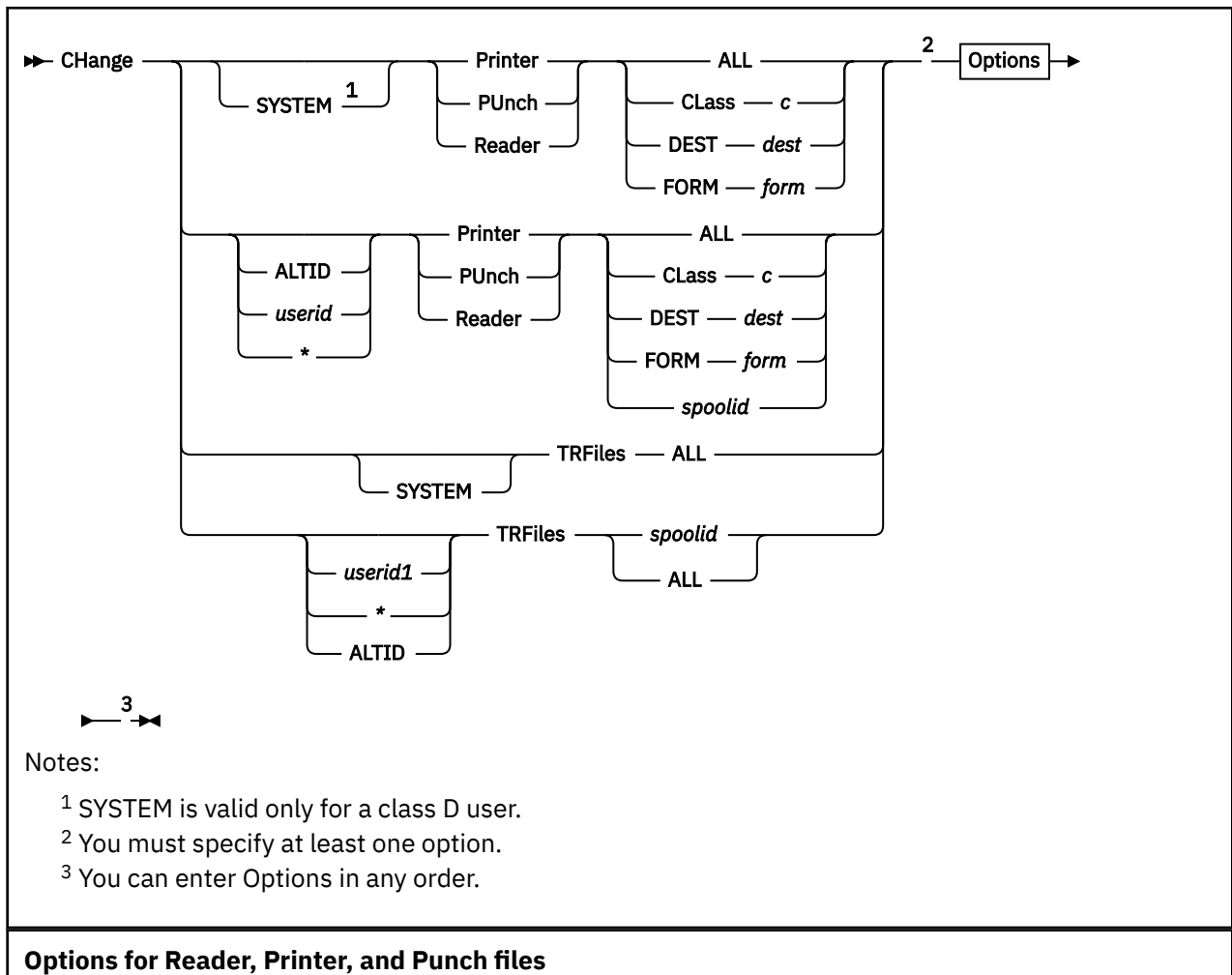


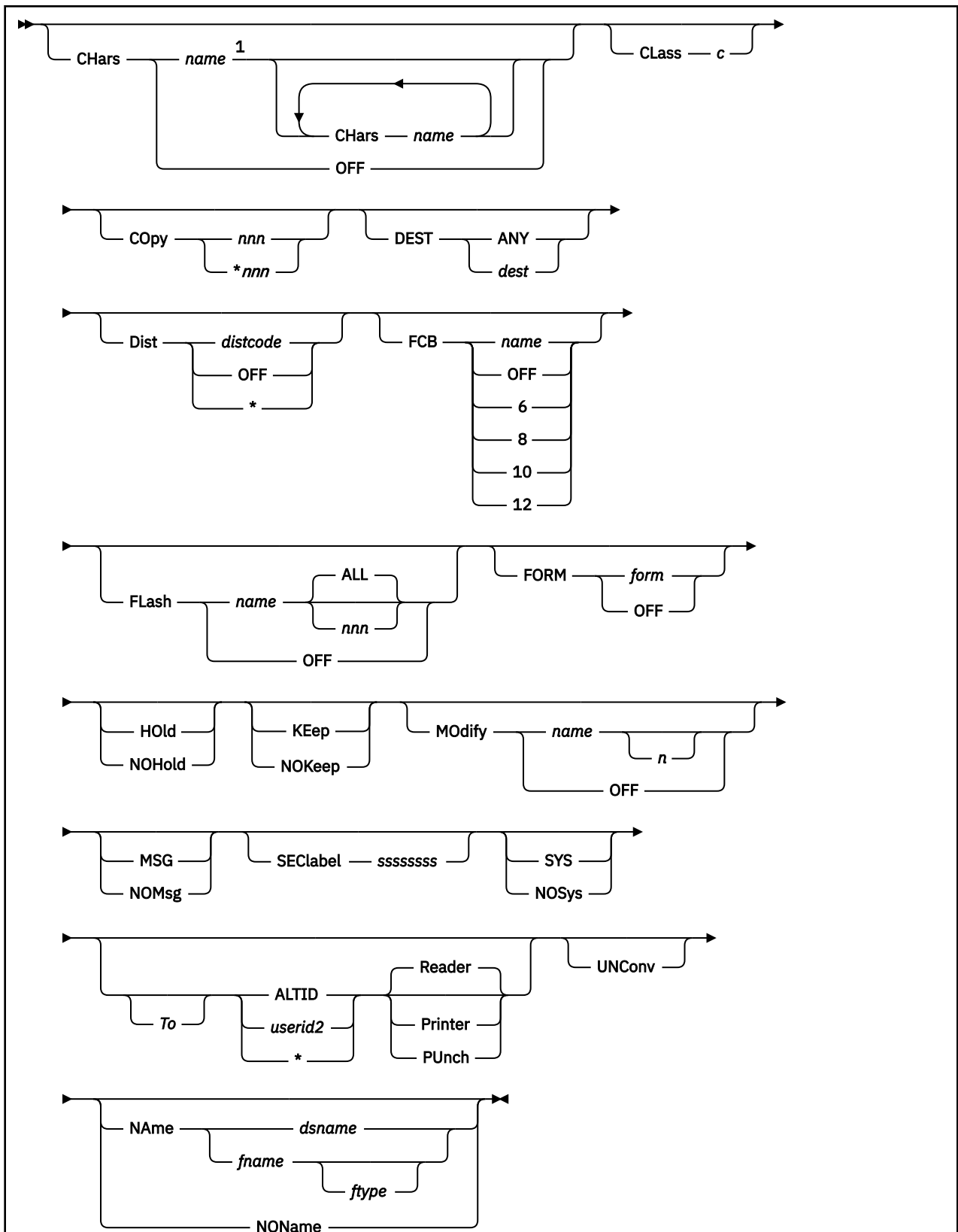
3. If you enter the BEGIN command and specify a starting storage location, execution begins at that address. The address you specify replaces the instruction address in the PSW. Then the PSW is loaded.
4. BEGIN terminates SLEEP status. For more information about SLEEP status, see the SLEEP command.
5. BEGIN cannot be used to resume execution on check-stopped processors. Refer to the SYSTEM RESET and SYSTEM CLEAR commands.
6. BEGIN cannot be used to resume execution on secondary processors that are not operational. In order for a secondary CPU to be started, the virtual machine must be in z/Architecture mode and the guest operating system must support secondary processors.

## Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP1005E An invalid character was encountered.
- HCP1454E CPU *nn* does not exist.
- HCP1455E CPU *nn* is not started because it is a secondary processor that is not operational.
- HCP1455E CPU *nn* is not started because it is in check-stop state.

## CHANGE





## Notes:

<sup>1</sup> You can specify up to four character set names.

## Options for system trace files (TRFILES)

## CHANGE



### Authorization

Privilege Class: D, G

### Purpose

Use CHANGE to alter the following:

- One or more external attributes of a closed spool file or files.
- User hold status of an active spool file.

Class D users can change the security label (SECLABEL) associated with any spool file in the system if the following conditions are met:

- An external security manager (ESM) is installed
- Security label checking is enabled, and:
  - The user is exempt from ESM authorization checking, or
  - The system has been placed in a tranquil state by the security administrator.

For additional information, contact your security administrator.

If you are a class D user, use CHANGE SYSTEM to change any spool files on the system, regardless of origin.

If you are a class G user, use CHANGE to manipulate files you own or have created. Though you cannot use the SYSTEM option, you can use the *userid* option to manipulate your own files, files owned by your alternate user ID, or files you have created and subsequently transferred.

### Operands

#### SYSTEM

allows the class D user to manipulate any files in the system, regardless of ownership. This is not allowed with the *spoolid* option.

#### *userid*

\*

#### ALTID

identifies the user whose spool files are to be changed. If you are changing your own files, you may specify your user ID as an asterisk (\*), which is also the default. ALTID indicates that the files belonging to your alternate user ID are to be changed.

#### Printer

##### PRT

changes printer spool files.

#### Punch

##### PCH

changes punch spool files.

#### Reader

##### RDR

changes reader spool files.

#### TRFiles

specifies that the spool file is to be directed to the system trace files (TRFILES) queue.

**ALL**

specifies that all files, regardless of their class, form, or *spoolid*, are to be changed.

**Class c**

designates a spool file class, where *c* is a 1-character alphanumeric field from A to Z or 0 to 9.

**DEST *dest***

specifies a 1- to 8-character name. Use *dest* to select those files to be changed.

**FORM *form***

designates a 1- to 8-character form name. Use *form* to select those files you want to change. For the Class G user, *form* is always the user form name. For the Class D user, *form* is the user form name when the *userid* is specified as an asterisk (\*) or the default user ID. Otherwise, *form* is the operator form number.

***spoolid***

identifies the file you are selecting. Note that *spoolid* is not compatible with the keyword SYSTEM.

**Options**

**Note:** You must choose at least one option. If you specify more than one, you can combine them in any order, except for the NAME option, which you must specify last.

**HOLD****NOHOLD**

HOLD puts a file in user hold status and prevents it from being punched, printed, or read. NOHOLD releases the specified spool files that are in user hold status.

**KEep****NOKeep**

KEEP retains a processed spool file in user hold status. NOKEEP resets files that are in keep status.

**MSG****NOMsg**

MSG controls whether messages are sent to the file owner when the file begins processing, and again when the file completes processing. Although you can enter these options for any type of spool file, no message is sent when a virtual reader processes the file. NOMSG resets the MSG option.

**SYS****NOSys**

SYS puts a file in system hold status and prevents it from being punched, printed, or read. NOSYS releases the specified spool file or files in system HOLD status. (User hold status can coexist with system hold status.)

**Note:** This option is available to class D users only.

**TRFiles**

changes system trace spool files.

**UNConv**

changes the spool file to an unconverted status, thereby canceling any preprocessing that the print server may have done on the file. Options for the file can now be changed, and the file may then be selected for conversion using the new options.

**Class c**

designates the new spool class, *c*, for the files.

**COpy *nnn*****COpy \**nnn***

specifies the number of file copies you want printed or punched when the file is produced. The value of *nnn* (number of copies) must be a decimal number from 1 through 255. Leading zeros need not be specified.

When you specify the optional asterisk (\*), the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

## CHANGE

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

### **DIST *distcode***

specifies the new distribution code for the spool files. The *distcode* variable is a 1- to 8-character alphanumeric identifier that appears on the output separator of punch or print files.

### **DIST OFF**

#### **DIST \***

resets the distribution code of the spool files to the default specified in your system directory entry.

### ***userid2***

\*

### **ALTID**

sends the spool file to the virtual machine specified by *userid2*. If no device type is specified immediately after *userid2*, the spool file is placed on the *userid2* reader queue. Specify an \* if you want to direct the output to your own virtual machine. ALTID indicates that files are to be directed to your alternate user ID.

### **Reader**

#### **RDR**

#### **Printer**

#### **PRT**

#### **PUnch**

#### **PCH**

specify, in conjunction with the TO operand, the file queue to which the spool file or files are to be directed.

### **FORM *form***

designates the new 1- to 8-character form name for the spool file. For the Class G user, *form* is the user form name. For the Class D user, *form* is the user form name when the *userid* is specified as an asterisk (\*) or the default user ID. Otherwise, *form* is the operator form number.

**Note:** When *form* is the user form name, both the user form name and the operator form number are changed.

### **FORM OFF**

resets the form for the spool files to the system default.

### **DEST *dest***

indicates the new destination value for the files being changed. The *dest* variable is a 1- to 8-character name specifying the output device or devices eligible to process the file.

### **DEST ANY**

indicates that the file can be processed on any CP output device that meets other selection criteria, regardless of the output device DEST settings.

### **Flash *name* ALL**

#### **Flash *name* *nnn***

signifies that a form overlay contained in the 3800 printer is to be superimposed (or flashed) onto the specified number of copies of the output. The *name* variable, 1- to 4-characters in length, identifies the forms overlay to be used in the projection. The number *nnn* must be in the range of 0 to 255, and it specifies that the first *nnn* copies of output are to be flashed.

For example, if you specify COPY 4 and FLASH *name* 2, the first two copies of your file are flashed, whereas the last two copies remain unflashed. If ALL is specified, the flash count always matches the copy count. ALL is the default.

### **Flash OFF**

resets the flash parameters. Blanks are inserted into the flash *name* field of the specified spool files, and the count is set to 0.

**MOdify name *n***

assigns a copy modification module to be used when printing the spool files on the 3800. This function makes it possible to alter text by either preventing the printing of certain information or by adding labels to output. The name must be one to four characters.

The *n* variable selects one of the character sets specified by the CHARS operand to be used for the copy modification text. The values 0, 1, 2, or 3 correspond to the order in which the character sets have been specified. If no *n* is specified, the first character set (0) is assumed.

**MOdify OFF**

resets the modify parameters. Blanks are inserted into the modify name field of the spool files and, when printing takes place, no modification occurs. The modify number (*n*) is set to 0.

**FCB *name*****FCB 6****FCB 8****FCB 10****FCB 12**

specifies the forms control buffer to be used when printing the file on the 3800 printer. The *name* variable is a 1- to 4-character name of an FCB image, or the number 6, 8, 10, or 12. The number tells the 3800 to print the entire spool file at 6, 8, 10, or 12 lines to the inch, regardless of the size of the paper currently loaded. Using these unique numbers assures that printed material fits onto a specified page size. Note, however, that FCB 10 prints only on a 3800 Model 3 printer.

**FCB OFF**

resets the FCB parameter. Blanks are inserted into the FCB *name* field of the spool file or files and, when the file is printed, a default FCB controls the vertical spacing of the output text.

**CHars *name*****CHars *name* CHars *name***

specifies the name of the character set or sets used to print the spool file or files on a 3800 printer. The name or names must be from one to four characters with a maximum of four names specified. You need to use CHARS more than once only if *nameN* is the same as an option name (for example, FORM) or with a user ID. If more than one name is specified, and modification is also specified, the order in which the names appear determines the table used for copy modification text.

**CHars OFF**

resets the CHARS parameters. Blanks are inserted into the name fields of the character sets used for output generation. The spool files, therefore, use a default character set when actually printed.

**SEClabel *sssssss***

assigns a security label to the chosen files. *sssssss* is the security label to be assigned to the files, and is specified as a 1- to 8-character alphanumeric value. The SECLABEL option is valid only for class D users.

**Name *fname ftype*****Name *dsname***

NAME indicates the name is to be assigned to the spool file for identification. For output files, this name appears on the separator page or cards.

Use *fname ftype* to name files in the CMS file name and file type format. Each of these fields is a 1- to 8-character alphanumeric name. If you omit the *ftype* field, blanks are substituted.

If you want to name files in a format suitable for MVS™ use (for example, SYS1.SYSLIB.MAC1), use *dsname*. This name can be up to 24 characters long; however, if *dsname* is more than 16 characters long, it is truncated.

If specified, the NAME option must be the last operand entered.

**NOName**

resets the NAME information to blanks.

**Responses**

Response 1:

## CHANGE

```
{nnnnnn} FILE(S) CHANGED  
{ NO }
```

indicates how many files have been changed. This response is sent to the user issuing the CHANGE command.

Response 2:

If you use the TO keyword to transfer a spool file to another user, the following response is sent to you and to the user receiving the file (if that user is active on the system):

### Note:

1. In the response sent to the operator receiving the file, *spd1* and *spd2* will be in the reverse order of that shown in the following response.
2. If an external security manager is installed on your system, this command will fail if you are not authorized to CHANGE the spool file TO the specified user. For additional information, contact your security administrator.

```
{RDR} FILE spd1 SENT {TO } userid {RDR} {AS } spd2 RECS {nnnn}...con't  
{PRT} {FROM} {PRT} {WAS} {nnnK}  
{PUN} {PUN} {nnnM}
```

Continued:

```
CPY [*]nnn c {HOLD } {KEEP }  
{NOHOLD} {NOKEEP}
```

### RDR

### PRT

### PUN

indicates the spool file queue. If you are transferring a file to another virtual machine, this field indicates the queue from which the file originated. If you are receiving a file from another virtual machine, this field indicates the file queue to which the file has been added.

### spd1

is the spool ID assigned when the command is entered.

### TO *userid*

indicates the user to whom the file has been sent.

### FROM *userid*

indicates the user from whom the file came.

### RDR

### PRT

### PUN

indicates the spool file queue. If you are transferring a file to another virtual machine, this field indicates the queue to which the file has been added. If you are receiving a file from another user, this field indicates the file queue in which the file originated.

### spd2

is the spool ID of the file when the command is complete.

### RECS *nnnn*

### RECS *nnnK*

### RECS *nnnM*

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

### CPY *nnn*

### CPY \**nnn*

specifies the number of file copies you want printed or punched when the file is produced.



When the optional asterisk (\*) is specified, the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

**c**

is the spool file class assigned to the file.

**HOLD**

**NOHOLD**

is the user HOLD status associated with the file.

**KEEP**

**NOKEEP**

is the KEEP status associated with the file.

Response 3:

```
{PRT} rdev {PRINTING} userid FILE spoolid RECS {nnnn}...con't
{PUN}      {PUNCHING}
Continued:
COPY [*]nnn SEQ sss CLc typ
```

is the message you receive when the system begins processing a spool file with the MSG attribute.

**PRT**

**PUN**

indicate the type of device on which the file is being processed.

**rdev**

is the real device number.

**userid**

is the name of the spool file owner.

**FILE spoolid**

is the spool file number.

**RECS nnnn**

**RECS nnnK**

**RECS nnnM**

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

**COPY nnn**

**COPY \*nnn**

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (\*) is specified, the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

**SEQ sss**

is the sequence number for the file on the printer or punch.

**CL c**

is the spool file class.

**typ**

shows the device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP, or SYS.

## CHANGE

Response 4:

```
SPOOLID spd1 HAS BEEN CHANGED to spd2
```

is the response you receive when a converted file is changed to unconverted status, a new owner, a new queue, or an option that affects the final printing of the file. CP assigns a new spool ID to the file and marks it as unconverted.

### **spd1**

is the spool ID assigned to the file before it was changed.

### **spd2**

is the new spool ID assigned to the file after it was changed.

## Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.
2. You can use QUERY RDR/PRT/PUN to determine the attributes of a spool file before and after you enter the CHANGE command.
3. You can change the status of an active file to HOLD in order to stop processing a file currently being printed or punched. No other attributes of an active file may be changed.
4. Class G users can change files they own or have created from their primary or alternate IDs.
5. If you use CHANGE to transfer a file to another user and you do not specify a distribution code, the distribution code is set to the receiving user's default distribution.
6. Although you can specify the CHANGE options of FLASH, FCB, MODIFY, CHARS, and COPY \* *nnn* for any spool file or files, they lead to system action only when a spool file with these attributes is printed on a CP-driven 3800 printer.
7. A user can cancel settings of the CHARS, MODIFY, or FCB options by specifying NULL in the name fields of these operands. In this case, NULL performs the same function as OFF.
8. The keyword OFF, which resets attributes to their default values, cannot also be used to specify a FORM, FLASH, FCB, MODIFY, DISTcode, or CHARS name. Similarly, you cannot use NULL to specify a CHARS, MODIFY, or FCB name.
9. If the NORETURN option was specified when the file was created by DIAGNOSE X'94', the originator of the file cannot retrieve that file from the current owner. If you try, the following message is issued:

```
FILE NOT CHANGED;  
FILE CREATED WITH THE NORETURN OPTION
```

10. When a file changes ownership, the spool file ID number also changes. The new number comes from the recipient's pool of available IDs. Thus, if user DORA uses the CHANGE command to send file #3 to user LILA, DORA gets the response:

```
RDR FILE 0003 SENT TO LILA  
RDR WAS 0018 RECS 1000 CPY  
001 A NOHOLD NOKEEP
```

LILA now has file #18.

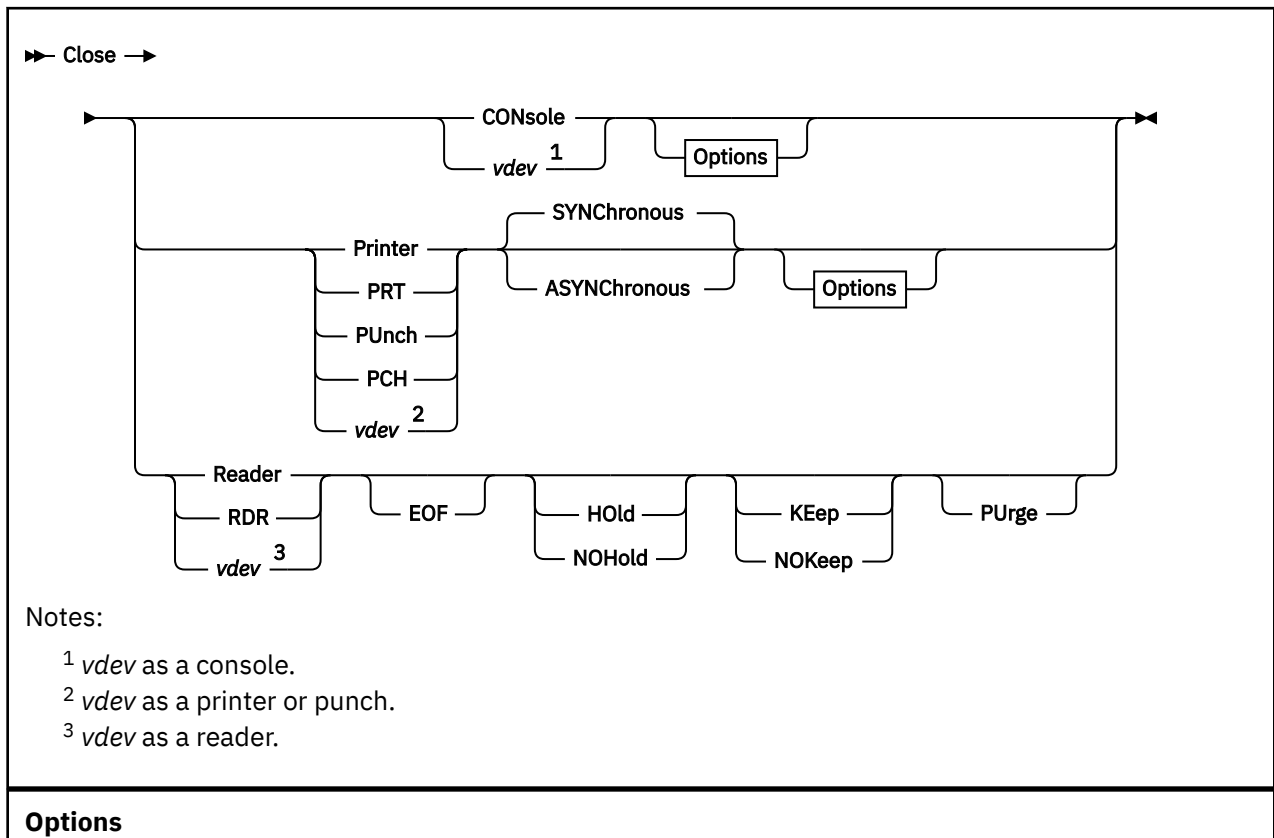
11. Most of the CHANGE command options which are valid for reader, printer, or punch files are not valid for system trace files (TRFILES). Refer to the CHANGE: OPTIONS syntax diagram above for the options which are valid for system trace files.
12. System trace files (TRFILES) may only be changed from one user's TRFILES queue to another user's TRFILES queue. Only system trace files may be changed to a TRFILES queue.
13. The SYS option puts a file in system hold and NOSYS releases the file from system hold. The SYS and NOSYS can be used in addition to HOLD and NOHOLD.

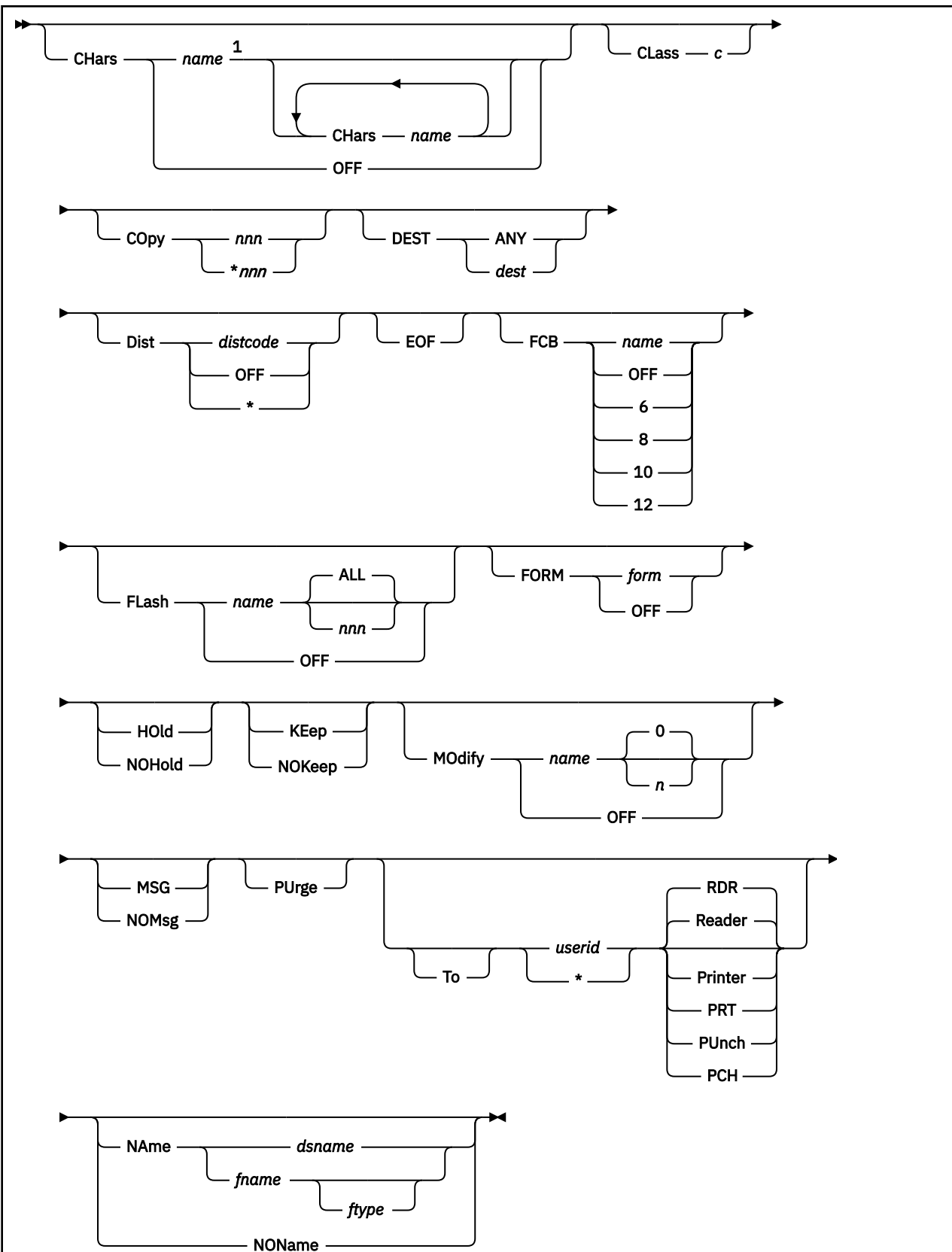
## Messages

- HCP002E Invalid operand - *operand*

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP028E Class missing or invalid
- HCP035E Device type missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP053E *userid/value* not in CP directory
- HCP088E An invalid number was supplied for {COPY|FLASH} - *operand*.
- HCP260E {Command|*command* command} processing cannot complete due to errors.
- HCP356E Access denied; User *userid* file *spoolid* not {changed|transferred|printed}
- HCP364E No alternate userid is currently set.
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP644E Unknown/unauthorized CP command: *command*
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*
- HCP1551E *value* is an invalid number for MODIFY.
- HCP1552E More than {4|8} operands were specified for the {CHARS|CLASS} option.
- HCP1559E The DSNAME is truncated to 16 characters.
- HCP1560E File was not {changed|transferred} because it was created with the NORETURN option.
- HCP2505E An output device is an invalid destination for system data files.
- HCP6525E External Security Manager is unavailable.

## CLOSE





Notes:

<sup>1</sup> You can specify up to four, four-character set names.

## Authorization

Privilege Class: G

## Purpose

Use CLOSE to end spooling on a virtual spool device (reader, printer, punch, or console).

## Operands

**CONsole**

**Printer**

**PRT**

**PUnch**

**PCH**

**Reader**

**RDR**

closes all reader, printer, punch, or console devices. Once a virtual console spool file is closed, it becomes a printer spool file and can be manipulated like any printer spool file.

**vdev**

identifies the virtual device number of the device to be closed. The device number may represent a console, reader, printer, or punch. *vdev* must be a hexadecimal number between X'0000' and X'FFFF'.

## Options

**Note:** You can specify one or more of these options. If you specify more than one, you can combine them in any order, except for the NAME option, which must be specified last.

**CHars *name***

**CHars *name* CHars *name***

specifies the name of the character set or sets used to print the spool file or files on a 3800 printer. The name or names must be from one to four characters long; as many as four names may be specified. You need to use CHARS more than once only if *nameN* is the same as an option name (for example, FORM) or a user ID. If more than one name is specified, and modification is also specified, the order in which the names appear determines the table used for copy modification text.

**CHars OFF**

resets the CHARS parameters. Blanks are inserted into the name fields of the character sets used for output generation. The spool files, therefore, use a default character set when actually printed on a 3800 printer.

**Class *c***

specifies the spool class to be assigned to the file, where *c* is a 1-character alphanumeric field from A to Z or 0 to 9.

**COpy \* *nnn***

specifies the number of file copies you want printed or punched when the file is produced. The value of *nnn* (number of copies) must be a decimal number from 1 to 255. You do not need to specify leading zeros.

When the optional asterisk (\*) is specified, the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

**DEST *dest***

overrides the destination value of the spool file. The *dest* variable is a 1- to 8-character name specifying the output device or devices eligible to process the file.

**DEST ANY**

sets the destination value to ANY. This allows the file to be processed on any CP output device that meets other selection criteria, regardless of the output device DEST settings.

**Dist *distcode***

specifies the distribution code to be assigned to the resulting spool file. The *distcode* variable is a one- to eight-character alphanumeric identifier that appears on the output printer/punch separator.

**Dist OFF****Dist \***

resets the distribution code for the resulting spool file to the default specified in the system directory for your user ID.

***userid***

\*

directs the spool file from the device to the virtual machine specified by *userid*. Specify an asterisk (\*) if you want to direct the file to your own virtual machine.

**EOF**

closes the spool file on the specified device regardless of the cont setting for the device. This option is not meaningful for a reader, it will be accepted but ignored.

**FCB *name*****FCB 6****FCB 8****FCB 10****FCB 12**

specifies the forms control buffer to be used when printing the file on a 3800 printer. The *name* variable is the one- to four-character name of an FCB image, or the number 6, 8, 10, or 12. This number tells the 3800 to print the entire spool file at 6, 8, 10, or 12 lines per inch, regardless of the size of the paper that is currently loaded. Note, however, that FCB 10 prints only on a 3800 Model 3 printer.

**FCB OFF**

resets the FCB parameter. Blanks are inserted into the FCB name field of the spool file and, when the file is printed on a 3800 printer, a default FCB controls the vertical spacing of the output text.

**Flash *name* ALL****Flash *name nnn***

signifies that a form overlay contained in the 3800 printer is to be superimposed (or flashed) onto the specified number of copies of the output. The *name* variable is one to four characters long, and identifies the forms overlay to be used in the projection. The *nnn* variable must be in the range of 0 to 255, and specify that the first *nnn* copies of output are to be flashed.

For example, if you specify COPY 4 and FLASH *name*2, the first two copies of your file are flashed; the last two copies remain unflashed. If ALL is specified, the flash count always matches the copy count. ALL is the default.

**Flash OFF**

resets the flash parameters. Blanks are inserted into the flash name field of the specified spool file, and the count is set to 0.

**FORM *form***

overrides the form name of a spool file. The *form* variable is one to eight characters long. If FORM is not given for the CLOSE command, the file created has a form specified by the last SPOOL command for the device, or a default name specified during system generation.

**FORM OFF**

resets the form to the default specified during system generation.

**HOLD****NOHold**

controls the user hold status.

## CLOSE

For output devices, HOLD prevents files from being punched or printed on a real device by putting a user hold status on the file. If you specify NOHOLD, you override the spool device HOLD option and prevent the file from being placed in hold status.

For input devices, the HOLD/NOHOLD options override the setting for the virtual device or the spool file. The file is not put in hold status unless the KEEP option is also specified. If you specify HOLD, the file is not purged after processing and remains eligible for the device. If you specify NOHOLD, the file is purged unless you also specify the KEEP option.

### **KEep**

#### **NOKeep**

controls the keep status, which indicates whether the current spool file is to be placed in user HOLD after processing.

For output files, the keep status is placed on the spool file. After it is printed or punched, the file is assigned user hold status and is not purged. The file's KEEP status is retained. Specify NOKEEP to override the current keep setting for the device so that keep is not set on the spool file.

For input files, KEEP overrides the keep setting on either the virtual reader or the spool file. After processing, the file is placed in hold status, and the KEEP option on the file remains unchanged. Specify NOKEEP to override either file or device keep characteristics. The disposition of a file closed NOKEEP (whether it is purged or retained) is controlled by the hold status of the virtual reader. If it were closed KEEP, it would never be purged.

### **MODify name n**

assigns a copy modification module to be used when printing the spool file on the 3800. This function makes it possible to alter text by not printing certain information or by adding labels to output. The *name* variable must be one to four characters long.

The *n* variable selects one of the character sets specified by the CHARS operand to be used for the copy modification text. The value of 0, 1, 2, or 3 corresponds to the order in which the character sets have been specified. If *n* is not specified, the first character set (0) is assumed.

### **MODify OFF**

resets the modify parameters. Blanks are inserted into the modify-name field of the spool file and, when the file is printed, no modification occurs. The modify number (*n*) is set to 0.

### **MSG**

#### **NOMsg**

indicates whether you want to be informed about file printing and punching. If you specify MSG, you are informed of file printing and punching. The message is sent to you when the file begins either printing or punching on a real device. Specify NOMSG to reset the MSG option.

### **NAme dsname**

#### **NAme fname ftype**

enables you to assign a name to the spool file for identification.

If you want to name files in a format suitable for MVS use (for example, SYS1.SYSLIB.MAC1), use *dsname*. This name can be up to 24 characters long; however, if *dsname* is more than 16 characters long, it is truncated.

The *fname ftype* specification permits you to name a file in CMS file name and file type format. The fields *fname* and *ftype* are one- to eight-character alphanumeric names. If you omit *ftype*, blanks are substituted.

If specified, the NAME option must be the last operand entered.

### **NOName**

resets the name information to blanks.

### **PURge**

closes and purges the spool file on the specified device regardless of the cont, keep, and hold setting for the device.



**Printer**  
**PRT**  
**PUnch**  
**PCH**  
**Reader**  
**RDR**

are used with the TO operand to specify the file queue to which the spool file is to be directed. If you do not specify a queue, the file is directed to the reader.

**SYNChronous**  
**ASYNChronous**

indicates whether the command issuer wishes to wait for the CLOSE command to complete.

If you specify ASYNCHRONOUS, you are free to process other commands while the CLOSE command completes. When it does, CP will notify you with a device end (DE) interrupt on the device affected.

If you specify SYNCHRONOUS, you are prevented from executing any other commands until the CLOSE command completes. When the command completes, the file will have been closed by CP.

## Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.
2. The options of the CLOSE command override the corresponding options in effect for the device you are closing. If you do not specify any optional operands, the options in effect for the device, or the status of the file you are processing, determines the file disposition.
3. Although the CLOSE options FLASH, FCB, MODIFY, and CHARS can be specified for any virtual device except a reader, they only lead to system action when a spool file with these attributes is printed on a CP-driven 3800 printer.
4. A user can cancel settings of the CHARS, MODIFY, or FCB options by specifying NULL in the name fields of these operands. In this case, NULL performs the same function as OFF.
5. The keyword OFF, which resets attributes to their default values, cannot also be used as the name specified in a FORM, FLASH, FCB, MODIFY, DISTcode, or CHARS operand. Similarly, NULL cannot be used to specify a CHARS, MODIFY, or FCB name.
6. If you close an output device (printer, punch, console) and the file you created is in hold, that file is not available for processing. If the file is not in hold, it is available for processing. You can transfer a file to another user by using the TO operand of the TRANSFER or CHANGE commands.
 

**Note:** An external security manager, if installed, may prevent you from transferring a spool file to another user unless you are specifically authorized to enter the TRANSFER and CHANGE commands. For additional information, contact your security administrator.
7. You can specify a file name and file type to help you identify a file and its contents. The file name and file type you assign have no effect on the way CMS names the file if it is read using the CMS READCARD or CMS DISK LOAD commands.
8. If you are closing a reader, the file that is being processed is purged unless HOLD or KEEP is in effect for the device or for the file. Use the appropriate CLOSE command option to override the effect of HOLD, NOHOLD, KEEP, or NOKEEP established for the reader or for the file.
9. If an input spool file has been only partially read (that is, your virtual machine did not get an end-of-file indication), you must enter a CLOSE READER command.
10. If the TO option is used to direct a file to a new user, the spool file ID number changes. The new number is obtained from the recipient's pool of available IDs. Thus, if user SAM enters CLOSE READER TO HARRY, he gets a response: RDR 0003 SENT TO HARRY RDR AS 0018. HARRY now has a file number 18.

If an external security manager is installed on your system, you may not be authorized to transfer a spool file to another user with the CLOSE TO command. For additional information, contact your security administrator.

## CLOSE

11. The disposition of a spool file that is to be processed on a real unit record device depends upon the resolution of HOLD, CONT, and KEEP status as implied by the device using the SPOOL and CLOSE commands.

For example, suppose your virtual printer is spooled CONT, and you enter `close prt class z`. Since you did not specify the CLOSE operand EOF, the file does not become available for processing on a real printer.

After completion of the CLOSE command, CP checks the hold status of the file to determine whether the files you closed can be processed. Thus, if hold is in effect for any of the closed files because you specified HOLD on the CLOSE command or because hold was set for the device using the SPOOL command, the file is not processed.

For readers only, the net effect of file status, device options, and the CLOSE command overrides is shown in [Table 11 on page 74](#).

Options in Effect for the Input File	System Action
NOKEEP/NOHOLD	The file is purged.
NOKEEP/HOLD	The file is not purged and remains available for processing in HOLD status.
KEEP/NOHOLD	The file is not purged. It is set in NOHOLD/KEEP status and is not available for processing.
KEEP/HOLD	The file is not purged. It is set in HOLD/KEEP status and is not available for processing.

In all cases, the original KEEP/NOKEEP status of the file is unchanged by the CLOSE operation.

Likewise, the disposition of a spool file that is to be processed on a real unit record device depends upon the resolution of HOLD, CONT, and KEEP status as implied by the device using the SPOOL and CLOSE commands. If an option is not specified on the CLOSE command, the option for the file is set from the corresponding option value on the virtual device.

12. If you use the ASYNCHRONOUS option when you issue the CLOSE command, CP will send a device end (DE) interrupt to the device affected when the command completes.
13. The CLOSE TO command may not be used to direct files to the RDR queue of a user with a password of NOLOG. CLOSE TO may be used to direct files to the PRT and PUN queues of a user with a password of NOLOG.

## Responses

Response 1:

```
{PRT}  rdev {PRINTING} userid
{PUN}      {PUNCHING}

FILE spoolid RECS {nnnn} COPY [*]nnn SEQ sss CL c typ
                {nnnK}
                {nnnM}
```

is the message you receive when the system begins processing a spool file with the MSG attribute.

**PRT**  
**PUN**

indicates the type of device on which the file is being processed.

**rdev**

identifies the real device number.

**userid**

identifies the user whose file was active.

**FILE spoolid**

identifies the unique, system-assigned spool ID number by which the system refers to the file.

**RECS nnnn****RECS nnnK****RECS nnnM**

identifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

**COPY \* nnn**

identifies the remaining number of copies of the file to be created.

When the optional asterisk (\*) is specified, the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

**SEQ sss**

identifies the sequence number for the file on the printer or punch.

**CL c**

is the spool file class.

**typ**

shows the device type that originally produced the spool file. The type can be RDR, PRT, CON, DMP, or SYS.

Response 2:

If you are using the TO *userid* option with the CLOSE command, the following response is sent to you and to the operator receiving the file (if that user is active on the system):

**Note:**

1. If an external security manager is installed on your system, you may not be authorized to transfer a spool file to another user with the CLOSE TO command. For additional information, contact your security administrator.
2. In the response sent to the operator receiving the file, *spd1* and *spd2* will be in the reverse order of that shown in the following response.

```
{RDR} FILE spd1 SENT {TO } userid {RDR} {AS } spd2
{PRT}                {FROM}      {PRT} {WAS}
{PUN}                {PUN}
{CON}
```

```
RECS {nnnn} CPY [*]nnn
c{HOLD } {KEEP }
  {nnnK}
  {nnnM}
  {NOHOLD} {NOKEEP}
```

**RDR****PRT****PUN****CON**

indicates the spool file queue. If you are transferring a file to another virtual machine, this field indicates the queue on which the file originated. If you are receiving a file from another virtual machine, this field indicates the file queue to which the file has been added.

**spd1**

identifies the spool ID of the file assigned when the command is issued.

## CLOSE

### **TO *userid***

indicates the user to whom the file has been sent.

### **FROM *userid***

indicates the user sending the file. If the PURGE option is used with the CLOSE command, this field is replaced by \*\*\*\* PURGED \*\*\*\* to indicate that the file has been deleted.

### **RDR**

### **PRT**

### **PUN**

indicates the spool file queue. If you are transferring a file to another virtual machine, this field indicates the queue to which the file has been added. If you are receiving a file from another virtual machine, this field indicates the file queue on which the file originated.

### ***spd2***

identifies the spool ID of the file after the command is complete.

### **RECS *nnnn***

### **RECS *nnnK***

### **RECS *nnnM***

identifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

### **CPY \* *nnn***

specifies the number of file copies you want printed or punched when the file is produced.

When you specify the optional asterisk (\*), the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

### **c**

identifies the spool file class assigned to the file.

### **HOLD**

### **NOHOLD**

identifies the user hold status associated with the file.

### **KEEP**

### **NOKEEP**

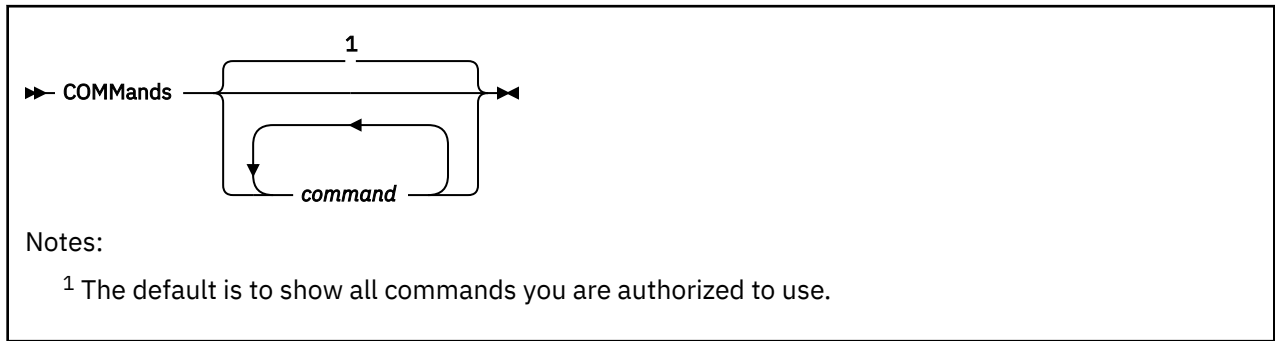
identifies the keep status associated with the file.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP030E Copies missing or invalid
- HCP035E Device type missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP052E Error in CP directory
- HCP053E *userid/value* not in CP directory
- HCP088E An invalid number was supplied for {COPY|FLASH} - *operand*.
- HCP418E {*rdev/vdev*} is not a spooled device.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.

- HCP427E *type {rdev/vdev/name[userid]}* system spool space full; file {closed|held|purged}
- HCP429E *type vdev* spool error; file {held|purged}
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*
- HCP1551E *value* is an invalid number for MODIFY.
- HCP1552E More than {4|8} operands were specified for the {CHARS|CLASS} option.
- HCP1558E Command processing cannot complete.
- HCP1559E The DSNAME is truncated to 16 characters.
- HCP6525E Authorization request failed.

# COMMANDS



## Authorization

Privilege Class: Any

## Purpose

Use COMMANDS to display the commands and diagnose codes you are authorized to use.

## Operands

### *command*

the names of commands to be checked for authorization.

## Usage Notes

1. The same function is available through the QUERY COMMANDS.
2. Some commands have more than one version, but the user may not be authorized to use all versions of the command. If the system rejects a command, it is probably because the user is not authorized to enter the command with the operand or operands entered.
3. To see the command versions you are authorized to use, enter `COMMANDS cmd`, where *cmd* is the name of the command or commands that interests you. You are shown the default IBM privilege classes of the commands that you are authorized to enter.

You may specify any number of commands. If a command name is entered twice, it appears twice in the response. No duplicate-entry checking is done.

4. Diagnose codes cannot be queried separately from commands.
5. `IBMCLASS=ANY` will output if:
  - the command is allowed before logon and not disabled, or
  - the user privilege class is ANY, meaning the command version is not associated with any particular privilege class and the command is enabled.

## Responses

Response 1:

```
cmd1    cmd2    cmd3    cmd4    cmd5    cmd6
cmd7...
:
diag1   diag2   diag3   diag4   diag5   diag6
diag7...
```

is the response when the command COMMANDS is entered alone; all authorized commands and diagnose codes are displayed.

**cmd1,2...**

the names of authorized commands.

**diag1,2...**

the names of authorized diagnose codes.

Response 2:

```
AUTHORIZED {QUERY|SET} OPERANDS:
operand1  IBMCLASS={c1,c2,c3}      operand2  IBMCLASS={c1,c2,c3}
           {ANY}                    {ANY}
           {NONE}                    {NONE}
operand3  IBMCLASS={c1,c2,c3}      operand4  IBMCLASS={c1,c2,c3}
           {ANY}                    {ANY}
           {NONE}                    {NONE}
:
```

is the response when the command COMMANDS is entered with the QUERY or SET operand; all authorized operands of the QUERY or SET command are displayed.

**operand1,2...**

the names of authorized QUERY or SET operands.

**c1,c2,...**

the IBM-defined privilege classes of the operands that you are authorized to enter.

Response 3:

```
cmd1      IBMCLASS=c1,c2,c3
cmd2      IBMCLASS=ANY
cmd3      IBMCLASS=NONE
```

is the response you receive when you specifically enter commands to check authorization for them. The response includes one line of output for each command listed.

**cmdx**

is the name of the command you entered on the command line.

**IBMCLASS=c1,c2,c3**

the IBM-defined privilege classes of the command specified; the privilege class displayed reflects the authorization of the user making the inquiry.

**IBMCLASS=ANY**

indicates that the command is a class ANY command, and all users may enter the command.

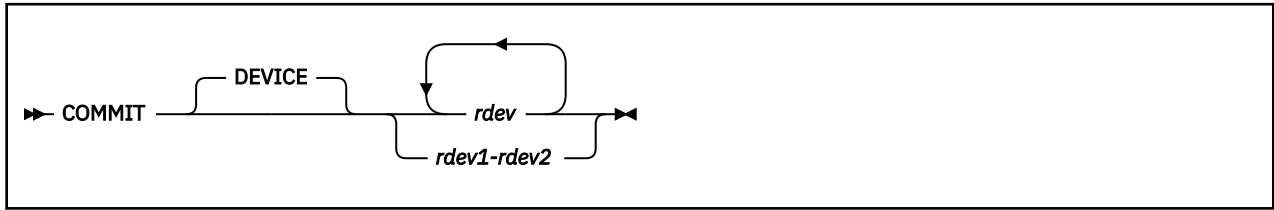
**IBMCLASS=NONE**

indicates that you are not authorized to enter any version of the command.

**Messages**

- HCP003E Invalid option - {option|command contains extra option(s) starting with option}

## COMMIT



### Authorization

Privilege Class: B

### Purpose

Use COMMIT to cause any cache fast write data to be transferred from the cache to the addressed device. This command is valid for cache storage subsystems that support the cache fast write function.

### Operands

#### DEVICE

emphasizes that only data for the specified devices is to be committed (for example, written to DASD). This is the default value.

#### *rdev*

#### *rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers, of the devices to which the commit order is to be sent.

### Usage Notes

1. This command may produce delayed responses. For more information, see [“Delayed Responses to CP Commands”](#) on page 7.
2. The COMMIT command is only valid for cache storage subsystems that support the cache fast write function. An error message is generated if the subsystem containing the specified device does not support this function.

### Responses

Response 1:

*Immediate Response:* If the transfer of cache fast write data occurs immediately, the following response is issued to the originator of the command. This response is repeated for each device specified on the command line.

```
rdev Cache fast write data has been committed.
```

#### *rdev*

is the real device number of a device for which cache fast write data has been transferred to the backing storage device.

*Delayed Response:* If the transfer of cache fast write data does not occur immediately, the following response is issued:

```
Command started: COMMIT rdev.
```

When the transfer of cache fast write data completes, the following responses are issued:



```
rdev Cache fast write data has been committed.
```

Under certain circumstances, it is possible that a cache storage subsystem will not notify CP when it has completed the transfer of the cache fast write data. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing is ended after generating the following response:

```
Command results lost: COMMIT rdev.
```

If errors occur during the asynchronous processing of the command that prevent the successful completion of the command, the following response is displayed:

```
Command failed: COMMIT rdev.
```

### **rdev**

is the real device number of a device for which cache fast write data has been transferred to the backing storage device.

*Response to the System Operator:* If the originator of the command is not the system operator, the following response is issued to the system operator when the transfer of cache fast write data completes. This response is also repeated for each specified device.

```
rdev Cache fast write data was committed by userid.
```

### **rdev**

is the real device number of a device for which cache fast write data has been transferred to the backing storage device.

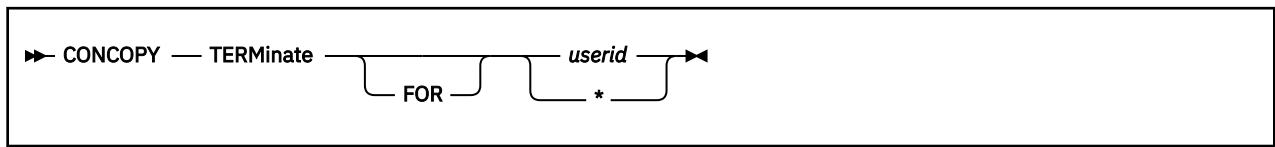
### **userid**

is the ID of the user who issued the command.

## **Messages**

- HCP006E Invalid device type - {rdev/vdev/ldev}
- HCP009E Invalid range - range
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP046E type rdev offline
- HCP319E Error processing subsystem command - cmd for DASD rdev
- HCP332E Invalid control unit type - rdev
- HCP1109E The command command cannot be processed for device rdev because the device is the secondary in a duplex pair.

## CONCOPY TERMINATE



### Authorization

Privilege Class: B

### Purpose

Use the CONCOPY TERMINATE command to end Concurrent Copy (CONCOPY) sessions on behalf of a virtual machine that started CONCOPY but which, because of a malfunction, cannot end CONCOPY itself. This command ends all the CONCOPY sessions on all storage controls for the virtual machine.

### Operands

#### *userid*

is the virtual machine for which CONCOPY sessions are to be terminated.

\*

indicates that CONCOPY sessions are to be terminated for the virtual machine issuing the command.

### Responses

Response 1:

```
All CONCOPY sessions have been terminated for userid
userid
```

indicates that the command has successfully ended all CONCOPY sessions for virtual machine *userid*.

Response 2:

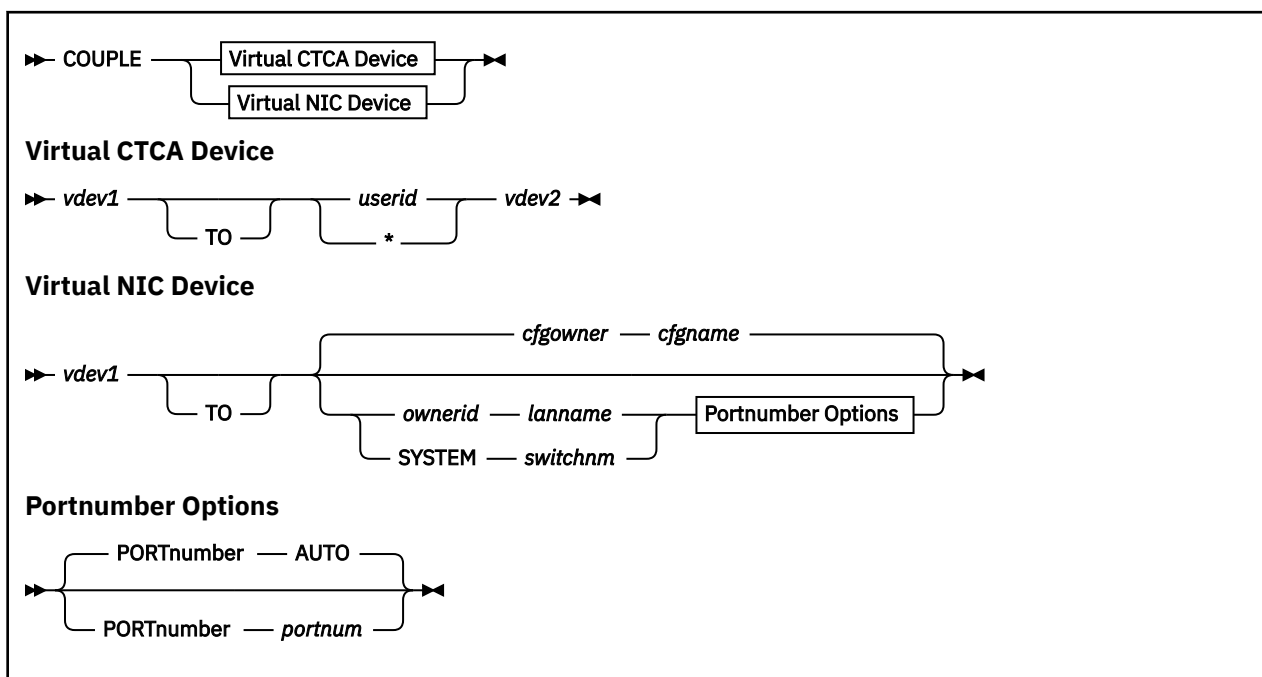
```
No CONCOPY session was found to be terminated for
userid userid
```

indicates that virtual machine *userid* does not have any CONCOPY sessions to be ended.

### Messages

- HCP026E Operand missing or invalid
- HCP319E Error processing subsystem command - *cmd* for userid *userid*

## COUPLE



### Authorization

Privilege Class: G

### Purpose

Use COUPLE to connect a virtual channel-to-channel adapter (CTCA) to a compatible virtual CTCA, or to connect a simulated network interface card (NIC) to a compatible guest LAN or virtual switch.

### Operands

#### **vdev1**

is the virtual device number (hex) of a CTCA, or network device in the invoker's virtual machine configuration. If *vdev1* is a virtual CTCA, it can be connected with a compatible virtual CTCA (designated by *userid vdev2*). If *vdev1* is a fully-simulated virtual OSA device type (created by the DEFINE NIC command), the adapter can be connected to a compatible guest LAN (designated by *ownerid* and *lanname*).

#### **userid vdev2**

is the target 'remote' virtual CTCA when *vdev1* is a virtual CTCA. If *userid* is specified as \*, it represents the user ID of the invoker. Use this form of the command to connect two virtual CTCA devices to emulate a single adapter. The invoker should be able to communicate with the 'remote' virtual machine using software appropriate for a compatible physical CTCA.

#### **cfghowner cfghname**

if a network is configured on the NICDEF user directory statement, the COUPLE command defaults to the configured network (*cfghowner cfghname*).

#### **ownerid lanname**

identifies a guest LAN as the target of the connection when *vdev1* is a virtual OSA device. The subject adapter is connected to the LAN named *lanname* owned by *ownerid*. Use this form of the command to connect a simulated network interface card (NIC) to a guest LAN. After a successful connection,

the invoker should be able to communicate with other virtual machines on the same guest LAN using software appropriate for a comparable physical network adapter.

#### **SYSTEM *switchnm***

identifies a virtual switch as the target of the connection when *vdev1* is a virtual OSA device. Use this form of the command to connect a simulated network interface card (NIC) to a virtual switch.

#### **PORTnumber *AUTO***

CP determines the virtual switch port number to be used.

Connect using an automatically assigned port number. If a NICDEF PORTNUMBER is configured for this device, the configured port and associated NICDEF attributes will be used. If a single port has been reserved for this user (using SET VSWITCH), the port and its associated port settings will be used. If more than one port has been reserved for this user, then the COUPLE command will fail because it cannot automatically resolve port number.

#### **PORTnumber *portnum***

The NIC is coupled to the specified *portnum* on the virtual switch. The specified *portnum* must be configured by SET VSWITCH, or the NICDEF must include network attributes. Otherwise the COUPLE command will fail. The *portnum* is a value from 1 to 2048.

### **Usage Notes**

1. If you do not specify TO, the *userid* you specify cannot be T or TO.
2. If *vdev1* is already connected to another virtual device, a guest LAN, or a virtual switch, that connection is broken. However, when a target virtual CTCA is already connected to another virtual device, the existing connection is **not** broken.
3. If an external security manager is installed on your system, you may not be authorized to enter this command. For additional information, contact your security administrator.
4. The connection established by the **COUPLE** command is automatically broken when the adapter, the guest LAN, or the virtual switch is detached. If you need to break this connection without detaching the adapter, the guest LAN, or the virtual switch, use the **UNCUPLE** command.
5. A simulated network interface card (NIC) can be coupled only to a compatible guest LAN or virtual switch. For example, a QDIO NIC can be coupled only to a QDIO LAN or a QDIO virtual switch. Any attempt to connect to an incompatible network will be rejected and reported by message HCP6024E.
6. To connect a NIC to a virtual switch:
  - The connection may be configured by the NICDEF user directory statement that defines this virtual NIC. This method binds the port configuration to a specific virtual NIC in the user directory. When NICDEF LAN is configured, it also authorizes the connection (without the need for a SET VSWITCH command). If the network is protected by an External Security Manager (ESM) then use the appropriate ESM interface to authorize this connection.
  - A user-defined port may be configured with the SET VSWITCH PORTNUMBER command. This method binds the network configuration to a specific port (in the range 1-2048) and allows the user to decide which virtual NIC to connect with each port. Use the appropriate External Security Manager (ESM) interface to grant authorization if the virtual switch is protected by an ESM.
  - A port may be configured with the SET VSWITCH GRANT command where the port number is assigned by z/VM. This method binds the network configuration to a user ID. Any virtual NIC belonging to this user will be connected to a port (in the range 2176-4095) and will adopt the port configuration. Use the appropriate External Security Manager (ESM) interface to grant authorization if the virtual switch is protected by an ESM.
7. After the virtual NIC is coupled to a virtual network, the port configuration can be changed dynamically:
  - Use a SET VSWITCH GRANT or PORTNUMBER command to change the configuration, including the configured Virtual Local Area Network (VLAN) ID set, dynamically. The guest does not have to uncouple and re-couple to activate this change. This method only overrides the NICDEF configuration until the next time this virtual NIC is coupled to the virtual network.

- If the NICDEF statement (in the user directory) provides network configuration, update the directory with a new configuration and refresh the configuration by having the guest uncouple and re-couple to the virtual network.
8. For a virtual CTCA the specified user must be logged on, and the virtual machine must have a virtual CTCA defined. The virtual machine may be operating in disconnect mode.
  9. If you want to connect the *vdev1* CTCA to another CTCA in your own virtual machine, specify *userid* as an asterisk (\*).
  10. The *vdev2* user must have authorized the connection using the DEFINE CTCA command or the SPECIAL directory entry.
  11. The originator of the command and the owner of the coupled CTCA (if different from the originator) receive messages indicating that the CTCAs have been coupled.
  12. It is possible to "UNCOUPLE" a virtual CTCA device by using an obscure form of the **COUPLE** command (**COUPLE vdev1 TO \* vdev1**). This method is supported for compatibility, but should be replaced by the **UNCOUPLE** command. Message HPCPCPL058E (RC58) results when using this method.
  13. The virtual ESCON® devices (defined with subclass SCTC, BCTC or CNC) cannot be coupled to virtual 3088 devices (defined with subclass CTCA or 3088), or to virtual FICON devices (defined with subclass FCTC). A virtual ESCON connection requires one controller (SCTC or BCTC) coupled with one non-control device (the CNC). The following combinations are permitted for ESCON devices:
    - COUPLE a subclass SCTC device with a subclass CNC device to get a virtual ESCON connection in Standard Mode
    - COUPLE a subclass BCTC device with a subclass CNC device to get a virtual ESCON connection in Basic Mode.

A virtual FICON device (defined with FCTC) cannot be coupled to virtual 3088 devices (defined with subclass CTCA or 3088) or to virtual ESCON devices (defined with subclass SCTC, BCTC, or CNC).

Any other combination of virtual device types will be rejected and the invoker will receive message HCP6024E.

## Responses

Response 1:

```
CTCA vdev1 COUPLED TO userid vdev2
```

is the normal response sent to the invoker of the COUPLE command when *vdev1* represents a virtual CTCA.

### **vdev1**

is the virtual device address of the invoker's CTCA.

### **userid vdev2**

is the user ID and virtual device address of the targeted CTCA.

Response 2:

```
CTCA vdev2 COUPLED BY invoker vdev1
```

is the normal response sent to the owner of the target CTCA.

### **vdev2**

is the virtual device address of the targeted CTCA.

### **invoker vdev1**

is the user ID and virtual device address of the originating CTCA.

Response 3:

```
CTCA vdev1 DROPPED FROM userid vdev2
```

## COUPLE

is the normal response sent to the invoker (and to the other userid) if the originating CTCA was already coupled to a different device.

### **vdev1**

is the virtual device address of the "local" CTCA.

### **userid vdev2**

is the user ID and virtual device address of the "remote" CTCA.

Response 4:

```
NIC vdev1 is connected to ownerid lanname
```

is the normal response sent to the invoker if *vdev1* is part of a simulated network interface card (NIC).

### **vdev1**

is the virtual device address that is part of a simulated network interface card (NIC).

### **ownerid lanname**

is the fully-qualified name of the coupled guest LAN.

Response 5:

```
NIC vdev1 is connected to SYSTEM switchnm
```

is the normal response sent to the invoker if *vdev1* is part of a simulated network interface card (NIC) connected to the virtual network via a port assigned by z/VM (2176-4095).

### **vdev1**

is the virtual device address that is part of a simulated network interface card (NIC).

### **switchnm**

is the name of the virtual switch.

Response 6:

```
NIC vdev1 is connected to SYSTEM switchnm port portnumber
```

is the normal response sent to the invoker if *vdev1* is part of a simulated network interface card (NIC) connected to the virtual network via a user-defined port (1-2048).

### **vdev1**

is the virtual device address that is part of a simulated network interface card (NIC).

### **switchnm**

is the name of the virtual switch.

### **portnumber**

is the port number assigned to the connection.

## Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP011E Invalid device type - *userid vdev*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP045E *userid* not logged on
- HCP047E *userid vdev* does not exist
- HCP058E CTCA *vdev* busy on *userid*
- HCP644E Unknown/unauthorized CP command *COUPLE*
- HCP2785E LAN name is missing
- HCP2786E NIC *vdev* not connected; the LAN name is not valid

- HCP2787E NIC *vdev* not connected; LAN *ownerid lanname* does not exist
- HCP2788E NIC *vdev* not connected; already connected to LAN *ownerid lanname*
- HCP2788E NIC *vdev* not connected; already connected to VSWITCH *switchnm*
- HCP2788E NIC *vdev* not connected; already connected to VSWITCH *ownerid switchname* port *portname*.
- HCP2789E NIC *vdev* not connected; LAN *ownerid lanname* has reached its MAXCONN limit
- HCP2791E NIC *vdev* was not connected to a LAN
- HCP2848E A VLAN list is not allowed for a [guest LAN | virtual switch with the VLAN UNAWARE attribute | virtual switch when your PORTTYPE is ACCESS]
- HCP2849E ESM VLAN configuration error. [No VLANs were provided for a VLAN AWARE virtual switch | Invalid VLAN ID provided.]
- HCP3034E NIC *vdev* not connected; port number was not specified.
- HCP3034E NIC *vdev* not connected; port *portnum* is not defined.
- HCP3034E NIC *vdev* not connected; NIC *vdev2* is already connected to VSWITCH SYSTEM *switchname* port *portnum*.
- HCP3034E NIC *vdev* not connected; Uplink NIC port number *portnum* does not match.
- HCP3034E NIC *vdev* not connected; Uplink NIC port number *portnum* was requested.
- HCP3034E NIC *vdev* not connected; port *portnum* is already connected.
- HCP3034E NIC *vdev* not connected; all system assigned ports are in use.
- HCP3041E Port number is missing.
- HCP3042E Port number is not valid.
- HCP3043E AUTO is the only PORTNUMBER option allowed for a guest LAN.
- HCP3044E ESM VLAN configuration error. ESM VLAN IDs do not match VLAN IDs for port *portnum*.
- HCP3046E Port number conflicts with configured port *portnum*.
- HCP3171E NIC *vdev* is not connected; COUPLE to an IVL virtual switch is not allowed.
- HCP6011E You are not authorized to COUPLE to {*userid vdev|ownerid lanname|ownerid switchname|ownerid switchname port portnum*}
- HCP6011E You are not authorized to COUPLE to {*userid vdev|ownerid lanname|ownerid switchname|ownerid switchname port portnum*} with NIC *vdev*.
- HCP6024E Incompatible device - *userid vaddr* is a *vdevtype* device.
- HCP6024E Incompatible LAN - *ownerid lanname* is a *vdevtype* LAN.
- HCP6525E External Security Manager is unavailable.
- HCP6528E The ESM has more than 2000 VLAN entries so the COUPLE command cannot be completed.

## CP

---



### Authorization

Privilege Class: Any

### Purpose

Use CP as a preface to identify control program (CP) commands issued.

### Operands

#### *command*

are the name and operands of the CP command or commands you enter. You must precede the first command with at least one blank.

### Usage Notes

1. In the CP environment, you can enter multiple commands as operands of the CP command, but you must separate each command by the logical-line-end (#) symbol.

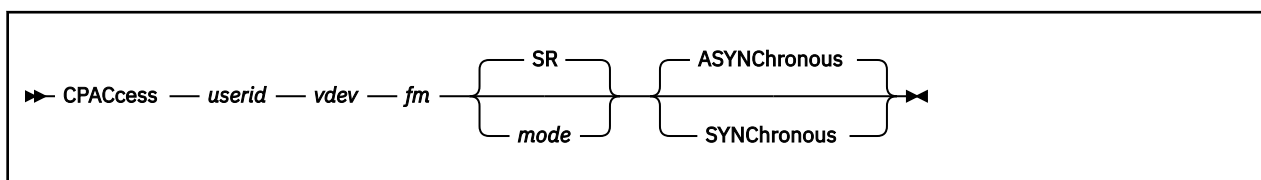
The pound sign (#) shown in the format block represents the logical-line-end symbol in effect for your virtual machine. If you have redefined the logical-line-end symbol, you must substitute your symbol for the pound sign (#).

Before you enter the CP command, be sure that your virtual machine is operating with the SET LINEDIT function set on. (This is the default.)

2. The CP command is treated as a null by the control program; for that reason it can precede any command if one or more blanks separates CP from the command.
3. If you enter CP with commands, you receive the responses appropriate to the commands entered.
4. The CP command, when not preceded with the logical line-end character, and when issued with the AT command, will not cause any virtual machine on the target system to enter the CP environment. However, the CP command can still be used to prefix other CP commands when issued with the AT command.



## CPACCESS



### Authorization

Privilege Class: A

### Purpose

Use CPACCESS to identify a CMS-formatted minidisk to CP and to make the files on that minidisk available to CP by establishing a file mode letter for the files. Before you can tell CP to link a minidisk, you must have the authority to access that minidisk. You do not have to actually link to or access the minidisk; you only need the capability to do so without specifying a password.

You can also do this by using the CP\_ACCESS statement in the system configuration file. For more information about the [CP\\_ACCESS Statement](#), see *z/VM: CP Planning and Administration*.

### Operands

#### *userid*

specifies the user ID of the owner of the minidisk that you want to make available to CP.

#### *vdev*

is the virtual device number of the specified user's minidisk, as defined in the user's entry in the system directory. The address can be any hexadecimal number between X'0000' and X'FFFF'.

#### *fm*

is the file mode letter that you want to assign to all files on the specified minidisk. You can specify any letter from A to Z.

#### *mode*

is the access mode. Valid modes are:

#### **ER**

Exclusive read-only access. CP establishes read access unless a user has an existing access to this disk. CP denies all requests for access to a disk with an existing exclusive mode.

#### **EW**

Exclusive write access. CP establishes write access unless a user has an existing access (any mode) to the disk. CP denies all requests for access to a disk with an existing exclusive mode access.

**Note:** An EXCLUSIVE access (ER, EW) is a stable access with the added restriction that no one else has, or can get access to, the specified minidisk until the access is released.

#### **M**

Multiple-write access. CP establishes write access unless a user has an existing write, stable, or exclusive mode access to the disk.

#### **MR**

Multiple-write access. CP establishes write access unless a user has an existing write, stable, or exclusive mode access to the disk. If a previous write or stable access exists, CP establishes read-only access. If an exclusive mode access exists, access is denied.

**MW**

Multiple-write access. CP establishes write access in all cases except when a user has an existing stable or exclusive mode access to the disk.

**R**

Read-only access. CP establishes read access unless a user has an existing write or exclusive mode access to the disk.

**RR**

Read-only access. CP establishes read access unless a user has an existing exclusive mode access.

**SM**

Stable multiple access. CP establishes write access unless a user holds a write, a stable, or an exclusive mode access to this disk. CP denies all requests for write access to a disk with an existing SM mode access.

**SR**

Stable read-only access. CP establishes read access unless a user has an existing write or exclusive mode access to this disk. CP denies all requests for write access to a disk with an existing SR mode access. SR is the default.

**SW**

Stable write access. CP establishes write access unless a user has an existing access (any mode) to this disk. CP denies all requests for write access to a disk with an existing SW mode access.

**Note:** A stable access means that the user holding the (SR, SW, or SM) access can be assured that the disk remains stable, unchanged by others, until the access is released.

**W**

Write access. CP establishes write access unless a user has an existing access (any mode) to the disk.

**WR**

Write access. CP establishes write access unless a user has an existing access (any mode) to the disk. If write access is denied, CP establishes read access unless another user has an existing exclusive mode access for the disk.

**ASYNChronous**

tells CP to schedule the command and lets you enter other commands before the CPACCESS command completes. When the CPACCESS command completes, CP displays a 1-line response, such as:

```
CPACCESS request for userid's vdev in mode mode completed
```

ASYNCHRONOUS is the default.

**SYNChronous**

tells CP to process the command immediately and does not allow you to enter any other commands until the CPACCESS command completes.

**Usage Notes**

1. Even though CP may establish write access to a minidisk (for example, by using the link mode MW), CP itself does not write to the minidisk.
2. When you enter the CPACCESS command and specify a file mode letter that CP is using for another minidisk, CP releases the first minidisk and accesses the minidisk you specified.
3. If the specified minidisk contains files that will be used frequently, you should either:
  - Issue the CPCACHE command to cache each file into storage.
  - Create a CPCACHE FILES file on the minidisk before you enter the CPACCESS command. The CPCACHE FILES file should contain the file names and file types of the frequently used files.

See [“CPCACHE” on page 94](#) for more information on caching files.

4. Use the CPRELEASE command to release a disk that you have accessed using the CPACCESS command. For more information, see [“CPRELEASE” on page 103](#).
5. Many users can link to a minidisk. We *strongly* recommend that you do not use the CPACCESS command to identify a minidisk to CP when some other user has write access to that minidisk. After accessing the minidisk, CP creates in storage a directory of all the files on that minidisk. If someone updates a file on that minidisk, CP is no longer maintaining a current directory and the results can be unpredictable. If you need to update a file on a disk that CP is accessing, use the CPRELEASE command to release the disk before you update the file.
6. CPACCESS accesses a minidisk for CP to read from. If you need to change a file on that disk, you must link and access the disk first. The size of the minidisk that can be accessed by the CPACCESS command is limited to 65520 cylinders.
7. If your virtual machine has privilege class A or E and can access a CP-accessed minidisk in read mode, you can use the CPTYPE command to look at the contents of a file on that disk. You do not have to link to the disk; you only need to have the capability to do so without a password. For more information, see [“CPTYPE” on page 105](#).
8. If the volume that contains the minidisk is under cross-system link protection (SSI cluster or XLINK), interpretation of the access mode takes into account all use of the minidisk by users on those systems.
9. The use of the stable and exclusive link modes (SR, SW, SM, ER, EW) is controlled by the LNKSTABL and LNKEXCLU options on the OPTION directory statement. For more information about the [OPTION Directory Statement](#), see *z/VM: CP Planning and Administration*.
10. If you have an ESM installed and the LINK command is controlled, the ESM will be queried to determine LINK authorization. For further information about linking restrictions, see [“LINK” on page 578](#).

## Responses

Response 1:

If you specify the SYNCHRONOUS operand, you see something like:

```
cpaccess oper1 06dc b sr synchronous
OPER1's 06DC accessed in mode B
Ready;
```

Response 2:

If you specify the ASYNCHRONOUS operand or you choose it as the default by specifying nothing, you see something like:

```
cpaccess oper1 06dc b sr
CPACCESS request for mode B scheduled
Ready;
CPACCESS request for OPER1's 06DC in mode B completed
```

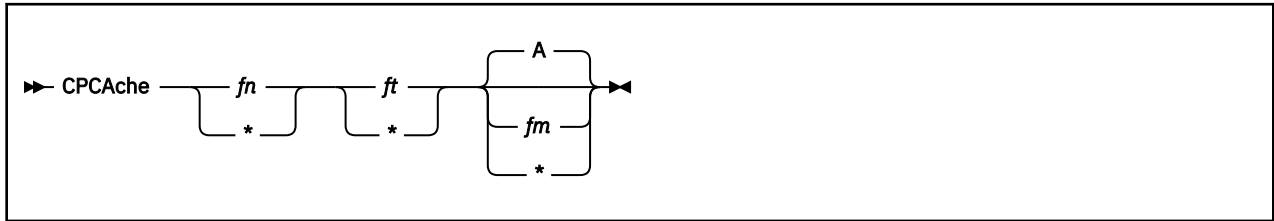
## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP007E Invalid userid - *userid*
- HCP046E DASD *rdev* offline
- HCP101E DASD *vdev* forced R/O; R/O by {*n* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[; stable by {*n* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP102E DASD *vdev* forced R/O; R/W by {*n* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[; stable by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]

- HCP103E DASD *vdev* forced R/O; R/W by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}, R/O by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; stable by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}
- HCP104E *userid vdev* not linked; R/O by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; stable by {*nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}|exclusive by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}
- HCP105E *userid vdev* not linked; R/W by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; stable by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}|exclusive by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}
- HCP106E *userid vdev* not linked; R/W by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}, R/O by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; stable by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}
- HCP107E *userid vdev* not linked; not in CP directory
- HCP108E *userid vdev* not linked; valid *valid* not mounted
- HCP109E *userid vdev* not linked; invalid link device
- HCP298E *userid vdev* not linked; request denied
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1008E Device *vdev* is not a DASD
- HCP1150E DASD {*rdev|vdev*} is not a valid base exposure.
- HCP2880E *userid vaddr* is not linked, CSE LINK I/O error, code *cc*.
- HCP6525E Authorization request failed.
- HCP6640E *userid vdev* not linked. DASD is a VDISK and is already linked on system *sysname*.
- HCP6640E *userid vdev* not linked. DASD is a VDISK and is defined under a SUBCONFIG for another SSI member.
- HCP6640E *userid vdev* not linked. MDISK DEVNO parameter conflicts with minidisk linked by {*userid|userid at sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk has been defined with the V mode suffix and is already linked by {*userid|userid at sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk has been defined with WRKALLEG and is already linked by {*userid|userid at sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk is already linked with the V mode suffix by {*userid|userid at sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk is already linked with WRKALLEG in effect by {*userid|userid at sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk is linked with the MDISK DEVNO parameter by {*userid|userid at sysname*}.
- HCP6640E *userid vdev* not linked. No EQID exists for RDEV *rdev*.
- HCP6640E *userid vdev* not linked. SSI cluster error, code *code*.
- HCP6645E *userid vdev* not linked; SSI cluster is in safe mode.
- HCP6648I MDISK *vdev* has virtual reserve/release support in effect.
- HCP6648I MDISK *vdev* has working allegiance simulation in effect.
- HCP6704E Missing token at end of line
- HCP6706E Invalid {device address|disk mode|access mode} - {*vdev|fm|mode|string*}
- HCP6731E {CPRELEASE|CPACCESS} request for mode *mode* already active.
- HCP6732I CPACCESS request for *userid*'s *vdev* in mode *fm* completed.
- HCP6733E Disk contains invalid control information -- disk not accessed

- HCP6733E Disk is not CMS EDF formatted minidisk -- disk not accessed
- HCP6733E I/O error encountered attempting to access disk -- disk not accessed
- HCP6733E Unable to link requested disk -- it cannot be accessed by CP
- HCP6735I Error encountered attempting to cache files -- *userid's vdev* accessed in mode *mode*
- HCP6772E Not authorized to link disk without a password

## CPCACHE



### Authorization

Privilege Class: A

### Purpose

Use CPCACHE to cause CP to cache a file on a CP-accessed minidisk. When you cache a file, CP moves the file into storage where it can find and access the file quickly. Use the CPCACHE command after you use the CPLISTFILE command to identify any files on the minidisk that are used frequently. After caching, you have faster access to your heavily-used files. You can also use a CP\_ACCESS statement in the system configuration file to identify a minidisk to CP. For more information on the CPACCESS command, see “CPACCESS” on page 89. You can use the CPCACHE command to identify frequently used files on minidisks that have been specified with a CP\_ACCESS statement. For more information on the CP\_ACCESS Statement, see *z/VM: CP Planning and Administration*.

### Operands

**fn**  
\*

is the file name of the file that you want CP to cache. If you specify an asterisk (\*), CP caches all file names. You can use generic file names to request a specific subset of files. A generic file name is a 1- to 8-character string with asterisks (\*) in place of 1 or more arbitrary characters and percent signs (%) in place of exactly 1 arbitrary character. For example:

```
cpcache hc%p* *
```

caches all files that start with HC and have P as their fourth character.

**ft**  
\*

is the file type of the file that you want CP to cache. If you specify an asterisk (\*), CP caches all file types. You can use generic file types to request a specific subset of files. A generic file type is a 1- to 8-character string with asterisks (\*) in place of 1 or more arbitrary characters and percent signs (%) in place of exactly 1 arbitrary character. For example,

```
cpcache * con%%g *
```

caches all files whose file types start with CON and have G as their sixth (and last) character.

**fm**  
\*

is the file mode letter of the minidisk that you had CP access. If omitted, the default is A. If you specify an asterisk (\*), CP caches the specified files on all accessed disks.

### Usage Notes

1. To release a minidisk and all cached files on that minidisk, enter a CPRELEASE command and specify the access mode of the minidisk. For more information, see “CPRELEASE” on page 103.

- If you find yourself caching the same files each time, you can avoid issuing the CPCACHE command by creating a CPCACHE FILES file on the minidisk. The CPCACHE FILES file lists all the files on that minidisk that you want CP to cache. You can use generic file names and file types. When you use the CPACCESS command or the CP\_ACCESS statement to identify the minidisk to CP, CP looks for a CPCACHE FILES file on that minidisk and caches all the specified files.

Here is an example of what a CPCACHE FILES file might look like:

```
/*-----
Cache often-used logo picture files
-----*/
TCP/IP LOGO
PVM LOGO

/*-----
Cache all log message files
-----*/
* LOGMS*
```

- If you want to see which files are being used the most, enter a CPLISTFILE command with the STAT option. This gives you a list of the files and how many times they have been opened and closed. For more information on the CPLISTFILE command, see [“CPLISTFILE” on page 99](#).

## Responses

Response 1:

If you want to cache all LOGO files on your A-disk that start with the string SYS, you can enter the following:

```
cpcache sys* logo a
1 file(s) cached
Ready;
```

Response 2:

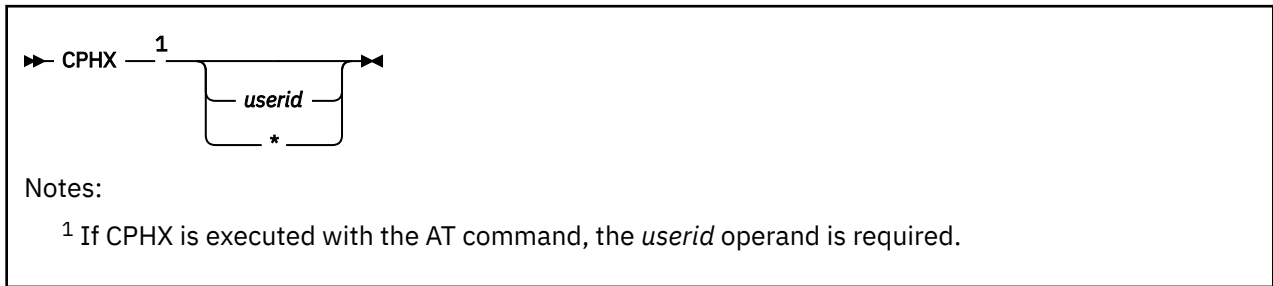
If CP cannot find any files that match the specified file name and file type, you see the following:

```
cpcache my logo a
No files cached
Ready;
```

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP6704E Missing token at end of line
- HCP6706E Invalid file {name|type|mode} - {fn|ft|fm}
- HCP6729I No files cached.
- HCP6731E Disk *fm* is being released -- command rejected

## CPHX



### Authorization

Privilege Class: A

### Purpose

Use CPHX to stop the execution of the following CP commands:

- ATTACH commands for devices
- DELETE EDEVICE command for devices
- FLASHCOPY commands for devices
- LOCATE commands searching storage
- LOCATEVM commands
- VARY ONLINE and VARY OFFLINE commands for devices
- VMDUMP command and DIAG94
- VMRELOCATE command with the SYNCHRONOUS option.

### Operands

*userid*

\*

is the user ID of the virtual machine whose command is to be canceled. An asterisk (\*) tells CP to use the issuer's user ID. You must have class A privilege to specify a user ID other than your own.

### Usage Notes

1. To cancel a command running in your own virtual machine, you must prefix the CPHX command with the string "#CP" followed by 1 blank, for example, #CP CPHX. The pound sign (#) represents the logical-line-end symbol in effect for your virtual machine. If you have redefined the logical-line-end symbol, you must substitute your symbol for the pound sign (#).
2. The ATTACH, DELETE EDEVICE, and VARY command processors check to see if a CPHX command has been issued before they process another device number specified on the ATTACH, DELETE EDEVICE, or VARY command.
3. The LOCATE and LOCATEVM command processors check to see if a CPHX command has been issued before they search another page of storage for data specified on the LOCATE or LOCATEVM commands.
4. The FLASHCOPY command processors check to see if a CPHX command has been issued before issuing additional I/O for device numbers that were specified on the FLASHCOPY command.
5. For the VMRELOCATE command with the SYNCHRONOUS option, CPHX does the same thing as the VMRELOCATE CANCEL operand. This is only applicable when the SYNCHRONOUS option was used to start the command. As with CANCEL, CPHX can only be used to stop the command up until the time that the request is sent from the source system to start the virtual machine on the destination system.



## Responses

Response 1:

If you use CPHX to stop a LOCATEVM command that is taking too long, you see something like this:

```
locatevm 0-end count all data C179022DF
The data is located at 00001270.
#cp cphx
HCPLMN6790I LOCATEVM processing halted. The next location
that would have been processed is 0001CF00.
Ready;
```

If you have a large virtual machine, the LOCATEVM command in the above example could take quite a long time.

Response 2:

If your user ID has class A privileges, you can use the CPHX command to cancel any of the previously mentioned CP commands issued by other virtual machines. For example, someone may have requested that a service machine attach a set of real devices:

```
sm rscs cp attach 0100-01ff to *
Ready;
```

An ATTACH command may take a long time because of hardware problems on a controller. If you notice that an ATTACH command is taking an unusually long time, you can choose to cancel the remainder of the ATTACH command. If you do, you see something like this:

```
cphx rscs
Your request to halt RSCS's ATTACH command has been initiated.
Ready;
HCPHXC6792I User RSCS's ATTACH command halted.
```

The message that informs you of the successful cancelation of the ATTACH command arrives some time after the CPHX command execution completes. The ATTACH command processing only recognizes the CPHX request when it begins to attach a new device.

The RSCS virtual machine's console would reflect the following activity as a result of the CPHX command execution:

```
09:01:52 Location YORKTOWN(MAINT) executing: CP ATTACH 0100-01FF TO *
:
HCPHXC6793I User MAINT has halted your ATTACH command.
HCPATR6790I ATTACH processing halted. The next device
that would have been processed is 0108.
```

The example above assumes that you are user ID MAINT. The last message indicates the ATTACH processing had completed processing devices 0100 to 0107 and was canceled before it tried to attach device 0108.

CP notifies the operator ID of the cancellation by issuing a message like this one to the operator's console:

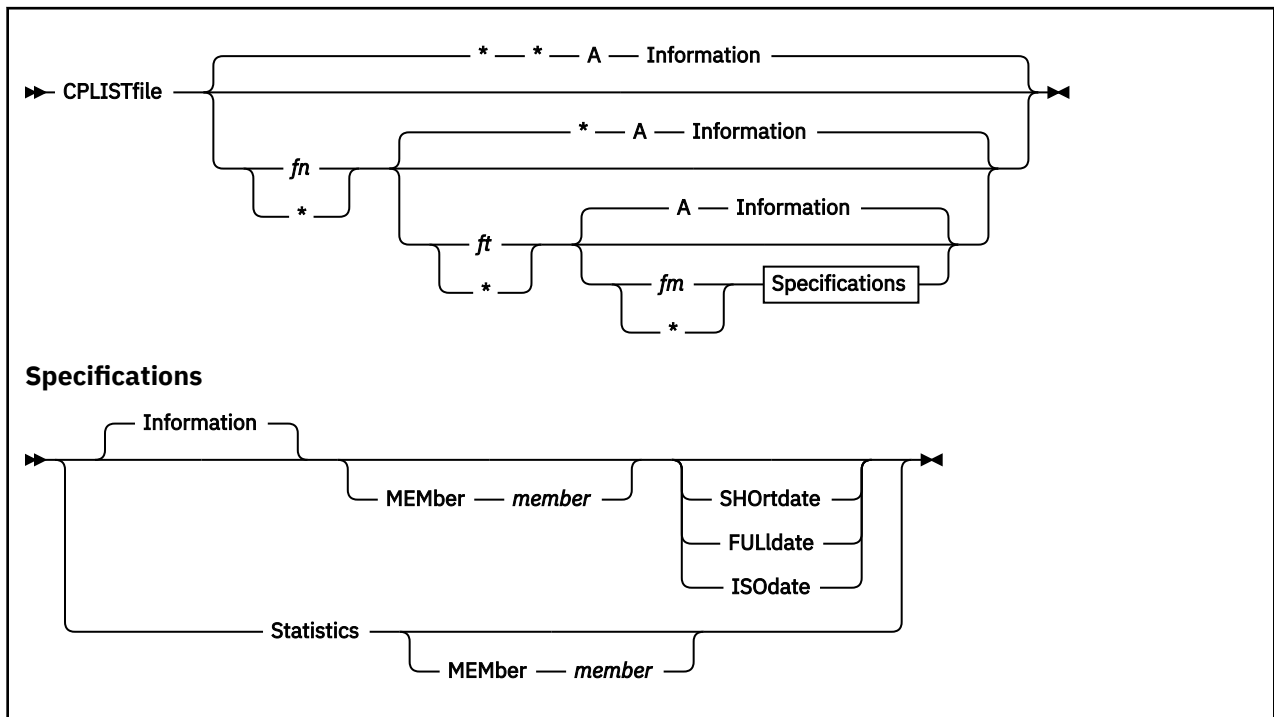
```
HCPHXC6791I User MAINT has halted the ATTACH command processed
by user RSCS.
```

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP644E Unknown/unauthorized CP command: *command*

- HCP6790I {ATTACH|DELETE EDEVICE|FLASHCOPY|VARY} processing halted. The next device that would have been processed is {*rdev*|*vdev*|*edev*}.
- HCP6790I {LOCATE|LOCATEVM} processing halted. The next location that would have been processed is *address*.
- HCP6790I {VMDUMP|DIAG94} processing halted.
- HCP6791I User *userid* has halted the *command* command executed by user *userid*.
- HCP6792I User *userid*'s *command* command halted.
- HCP6793I User *userid* has halted your *command* command.
- HCP6898E User *userid* is not executing a command that can be halted.
- HCP6899E A halt request for user *userid* has already been made by user *userid*.

## CPLISTFILE



### Authorization

Privilege Class: A

### Purpose

Use CPLISTFILE to list information on the files on a minidisk that you identified to CP by using either a CPACCESS command or a CP\_ACCESS statement.

For more information on the CPACCESS command, see “CPACCESS” on page 89.

For more information on the [CP\\_ACCESS Statement](#), see *z/VM: CP Planning and Administration*.

### Operands

**fn**  
\*

is the file name of the file on which you want information. If you omit *fn* or specify an asterisk (\*), CP lists all file names. You can use generic file names to request a specific subset of files. A generic file name is a 1- to 8-character string with asterisks (\*) in place of 1 or more characters and percent signs (%) in place of exactly 1 character. For example:

```
cpllistfile hc%p* * *
```

lists all files that start with HC and have P as their fourth character.

**ft**  
\*

is the file type of the file on which you want information. If you omit *ft* or specify an asterisk (\*), CP lists all file types. You can use generic file types to request a specific subset of files. A generic file type

is a 1- to 8-character string with asterisks (\*) in place of 1 or more arbitrary characters and percent signs (%) in place of exactly 1 arbitrary character. For example:

```
cplistfile * con%%g *
```

caches all files whose file types start with CON and have G as their sixth (and last) character.

### ***fm*** **\***

is the file mode letter of the minidisk containing the files on which you want information. If you omit *fm*, CP lists the files on the minidisk that is accessed as file mode A (the default). If you specify an asterisk (\*), CP lists the files on all accessed disks.

### **Information**

tells CP to list the following information on each file in the list:

- File ID (file name, file type, and file mode)
- Format (fixed or variable)
- Logical record length
- Number of records in the file
- Date and time when the file was created or last updated
- Whether CP cached the file.

INFORMATION is the default.

### **Statistics**

tells CP to list the following information about each file:

- File ID (file name, file type, and file mode)
- Number of times the file has been opened and closed since you identified the minidisk to CP using the CPACCESS command (see “CPACCESS” on page 89).

**Note:** If CP cached the file, the number of times the file has been opened is one more than the number of times the file has been closed. For example, see the PVM and TCPIP LOGO files in the two responses that follow.

### **MEMBER *member***

tells CP that the files you are displaying are members of a CMS partitioned data set. The variable *member* is the name of the member within the CMS partitioned data set that you want displayed, and must be a 1- to 8-character alphanumeric string.

If you omit the MEMBER operand, CP does not display any of the members of the CMS partitioned data set. CP only displays the name of the CMS partitioned data set.

You can use generic member names to request a specific subset of members. A generic member name is a 1- to 8-character string with asterisks (\*) in place of 1 or more characters and percent signs (%) in place of exactly 1 character. For example:

```
cplistfile * * a information member qu%r*
```

lists all members that start with QU and have R as their fourth character.

### **SHOrtdate**

Specifies that dates in the response to this command will be displayed in mm/dd/yy format, where mm is the month, dd is the day of the month, and yy is the 2-digit year.

### **FULldate**

Specifies that dates in the response to this command will be displayed in mm/dd/yyyy, where mm is the month, dd is the day of the month, and yyyy is the 4-digit year.

### **ISOdate**

Specifies that dates in the response to this command will be displayed in yyyy-mm-dd format, where yyyy is the 4-digit year, mm is the month, and dd is the day of the month.

## Usage Notes

1. To see a list of the minidisks that CP has access to, use the QUERY CPDISK command. For more information, see [“QUERY CPDISKS” on page 824](#).
2. If no date operand is specified, the user's default date format setting will be used.

## Responses

Response 1:

If you want to see a list of all the LOGO files on CP's A-disk, enter the following:

```
cplistfile * logo a information fulldate
Filename Filetype FM Fmt LRecl   Records   Date       Time       Cache
DAVID    LOGO      A  F    78      15 09/05/1992 06:43:26 No
DEFAULT  LOGO      A  F    80      15 11/29/1992 06:42:14 No
EMILY    LOGO      A  F    80      15 02/14/1992 06:42:23 No
LDEV     LOGO      A  F    80      15 07/28/1992 06:41:48 No
LOCAL    LOGO      A  F    80      15 10/03/1992 06:41:40 No
MOD5     LOGO      A  F   132     15 10/03/1992 06:41:15 No
PVM      LOGO      A  F    78      15 11/29/1992 06:42:01 Yes
SNA      LOGO      A  F    78      15 09/05/1992 06:41:34 No
SVCDIR   LOGO      A  F    80      15 12/18/1992 22:06:27 No
SYSTEM   LOGO      A  F    80      15 11/29/1992 06:41:28 No
TCP/IP   LOGO      A  F    78      15 10/03/1992 06:42:07 Yes
08E4     LOGO      A  F    80      15 12/18/1992 06:41:21 No
08E6     LOGO      A  F    80      23 12/18/1992 06:41:54 No
Ready;
```

Response 2:

If you want to see a list of all the LOGO files on CP's A-disk and you want to know how many times each file has been opened and closed, enter the following:

```
cplistfile * logo a statistics
Filename Filetype FM      Opens    Closes
DAVID    LOGO      A         0         0
DEFAULT  LOGO      A         1         1
EMILY    LOGO      A         0         0
LDEV     LOGO      A         0         0
LOCAL    LOGO      A         1         1
MOD5     LOGO      A         2         2
PVM      LOGO      A        19        18
SNA      LOGO      A         1         1
SVCDIR   LOGO      A         1         1
SYSTEM   LOGO      A         0         0
TCP/IP   LOGO      A        21        20
08E4     LOGO      A         0         0
08E6     LOGO      A         0         0
Ready;
```

Response 3:

If you want to see a list of all the TXTLIB files on CP's A-disk that contain CMS partitioned data set members starting with the prefix IEA, enter the following:

```
cplistfile * txtlib a information member iea* shortdate
Filename Filetype FM Fmt LRecl   Records   Date       Time       Cache
CMSLIB   TXTLIB   B  F    80      131 12/20/96 13:12:29 No
  Member IEAXPSIM starts at record 2
  Member IEAXPALL starts at record 8
  Alias  IEAXKALL starts at record 8
  Member IEAXPDXR starts at record 102
  Alias  IEAXKDXR starts at record 102
Ready;
```

This response tells you that CP has one TXTLIB on its A-disk and that TXTLIB contains:

- 3 members whose names start with IEA at records 2, 8, and 102
- 2 aliases whose names starts with IEA for the two members at records 8 and 102.

Response 4:

If you want to see the above response with the date in mm/dd/yyyy format, enter the following:

```
cplistfile * txtlib a information member iea* fulldate
Filename Filetype FM Fmt LRecl   Records   Date       Time    Cache
CMSLIB   TXTLIB   B  F      80        131 12/20/1996 13:12:29 No
  Member IEAXPSIM starts at record 2
  Member IEAXPALL starts at record 8
  Alias  IEAXKALL starts at record 8
  Member IEAXPDXR starts at record 102
  Alias  IEAXKDXR starts at record 102
Ready;
```

Response 5:

If you want to see the above response with the date in yyyy-mm-dd format, enter the following:

```
cplistfile * txtlib a information member iea* isodate
Filename Filetype FM Fmt LRecl   Records   Date       Time    Cache
CMSLIB   TXTLIB   B  F      80        131 1996-12-20 13:12:29 No
  Member IEAXPSIM starts at record 2
  Member IEAXPALL starts at record 8
  Alias  IEAXKALL starts at record 8
  Member IEAXPDXR starts at record 102
  Alias  IEAXKDXR starts at record 102
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP013E Conflicting option - *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid file {name|type|mode|member name} - {*fn* | *ft* | *fm* | *member*}
- HCP6728E {File|Member} not found.

# CPRELEASE



## Authorization

Privilege Class: A

## Purpose

Use CPRELEASE to release a disk that was previously accessed using the CPACCESS command or the CP\_ACCESS statement.

For more information on the CPACCESS command, see “CPACCESS” on page 89. For more information on the CP\_ACCESS Statement, see *z/VM: CP Planning and Administration*.

## Operands

### *fm*

is the file mode letter of the minidisk that you want CP to release.

### ASYNChronous

tells CP to schedule the command and lets you enter other commands before the CPRELEASE command completes. When the command is complete, CP displays the following:

```
CPRELEASE request for disk filemode completed
```

ASYNCHRONOUS is the default.

### SYNChronous

tells CP to process the command immediately and does not allow you to enter any other commands until the CPRELEASE command completes. CP must wait until all the files are closed before it can release a minidisk. If there are a large number of open files and you specify the SYNCHRONOUS option, you may be forced to wait for some time before you can enter any other commands.

## Usage Notes

1. When you enter a CPRELEASE command, CP removes from storage all the files on the minidisk that were cached into storage as a result of CPCACHE commands or a CPCACHE FILES file.

## Responses

Response 1:

If you specify the SYNCHRONOUS operand, you see something like this:

```
cprelease b synchronous
Disk B released
Ready;
```

Response 2:

If you specify the ASYNCHRONOUS operand or you choose it as the default by specifying nothing, you see something such as this:

## CPRELEASE

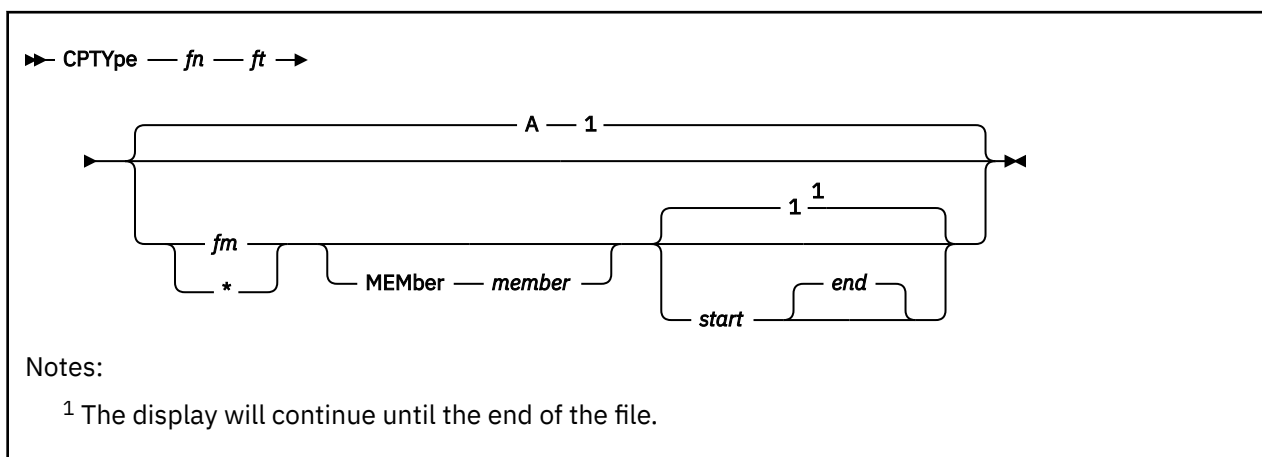
```
cprelease b asynchronous  
CPRELEASE request for disk B scheduled  
Ready;  
CPRELEASE request for disk B completed
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP6704E Missing token at end of line
- HCP6706E Invalid access mode - *string*
- HCP6706E Invalid disk mode - *filemode*
- HCP6731E CPRELEASE request for mode *filemode* already active



## CPTYPE



### Authorization

Privilege Class: A

### Purpose

Use CPTYPE to cause CP to display selected lines of a file on a minidisk that you identified to CP with either a CPACCESS command or a CP\_ACCESS statement.

For more information on the CPACCESS command, see “CPACCESS” on page 89. For more information on the CP\_ACCESS Statement, see *z/VM: CP Planning and Administration*.

### Operands

#### *fn*

is the file name of the file that you want to display.

#### *ft*

is the file type of the file that you want to display.

#### *fm*

is the file mode letter of the minidisk that you identified to CP by using the CPACCESS command or the CP\_ACCESS statement. If you omit *fm*, CP displays the specified file on the disk it has accessed as A (the default). If you specify an asterisk (\*), CP searches all the accessed minidisks and displays the first file found in the search hierarchy.

#### **MEMber** *member*

is the name of a member within the file that you want to display. Member is meaningful if your file is a CMS partitioned data set (for example, a TXTLIB), because this is the type of file that has members.

#### *start*

is the line number of the first record you want CP to display. You can specify any positive decimal number between 1 and the line number of the last record in the file. If this number is greater than the line number of the last record in the file, you receive an error message. If you do not specify a number, CP starts displaying the file at line 1 (the default).

#### *end*

is the line number of the last record you want CP to display. You can specify any positive decimal number between 1 and the line number of the last record in the file. If you do not specify a number, CP displays the file at the line specified by *start* and continues displaying until it reaches the end of the file.

## Usage Notes

1. If you specify the same line number for *start* and *end*, CP displays only that line.
2. CP limits the length of each output line to 130 characters.
3. Before you can use the CPTYPE command to display selected lines of a file on a minidisk, someone must have already issued a CPACCESS command or specified a CP\_ACCESS statement to identify the minidisk to CP. Although you do not have to have a link to that minidisk, your virtual machine must be capable of establishing one without a password. For more information on the CPACCESS command, see [“CPACCESS” on page 89](#). For more information on the [CP\\_ACCESS Statement](#), see *z/VM: CP Planning and Administration*.
4. To display a list of the minidisks that CP has access to, use the QUERY CPDISKS command. For more information, see [“QUERY CPDISKS” on page 824](#).

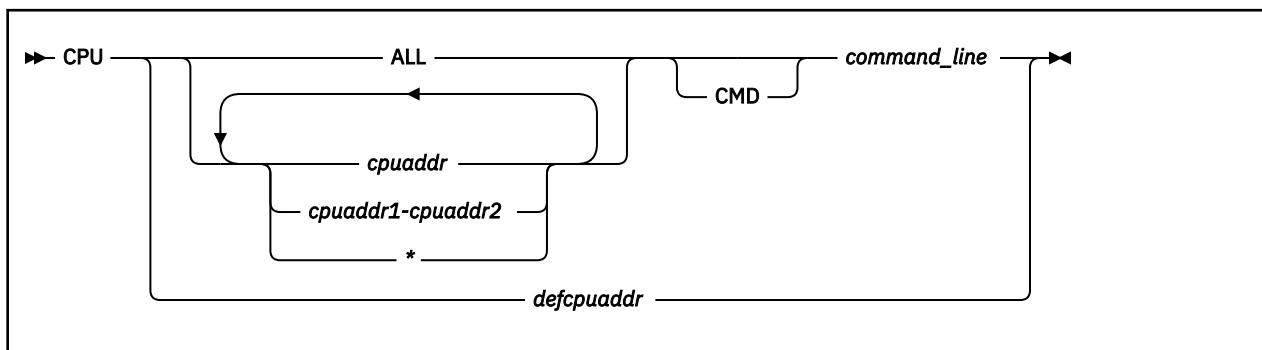
## Responses

CP displays the file on your terminal screen according to the information you specify.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP2768E Missing file member name
- HCP6702E Error encountered while attempting to read records from *fn ft fm*
- HCP6703E File *fn ft fm* [member *name*] not found
- HCP6704E Missing token at end of line
- HCP6706E Invalid file member name - *member*
- HCP6706E Invalid file mode - *fm*
- HCP6706E Invalid file name - *filename*
- HCP6706E Invalid file type - *ft*
- HCP6772E Not authorized to link disk without a password

## CPU



### Authorization

Privilege Class: G

### Purpose

Use CPU to do the following:

- Designate the virtual processor or processors to process the CP command specified
- Choose the default processor to run CP commands entered in response to CP read or prefixed by #CP.

### Operands

#### ALL

specifies that *command\_line* be run on all virtual processors currently defined in the principal configuration.

#### *cpuaddr*

identifies the address of the processor to run *command\_line*.

#### *cpuaddr1-cpuaddr2*

identifies a range of consecutive CPU addresses, designated by a nonwrapping pair of addresses separated by a hyphen (-) and not separated by blanks.

#### \*

specifies that the CP command requested should be run on the default target CPU. This option is useful to direct to the principal configuration a command issued from the adjunct configuration or issued via #CP while the adjunct is active.

The command can be directed to multiple CPUs using any combination of individual CPU addresses, ranges, and asterisk, separated by blanks. Commands are executed in the order the CPU addresses are listed. If a CPU address is included multiple times in the list, the subsequent references are ignored.

#### CMD

must be specified if CP would interpret the first token of the command line as a hexadecimal address.

For example, if you wish to specify that the DISPLAY PROGRAM STATUS WORDS command be processed on the CPU at address 1, do *not* specify *cpu 1 d psw* because CP would interpret the *d* as an address. Instead, enter *cpu 1 cmd d psw*.

#### *command\_line*

identifies the CP command you want to process.

When processing the CPU command, CP does not check the validity of the command entered with it. Only the processor receiving the CP command checks it for validity and errors.

### ***defcpuaddr***

identifies the address of the processor in the principal configuration that you want to process future commands issued from the console, unless they are overridden with a CPU command prefix. This is referred to as the default target CPU.

### **Usage Notes**

1. If *command\_line* is specified, it applies only to the specified virtual processors. If *defcpuaddr* is the only operand specified, future CP commands apply to that virtual processor until the next CPU command is issued.

### **Responses**

Response 1:

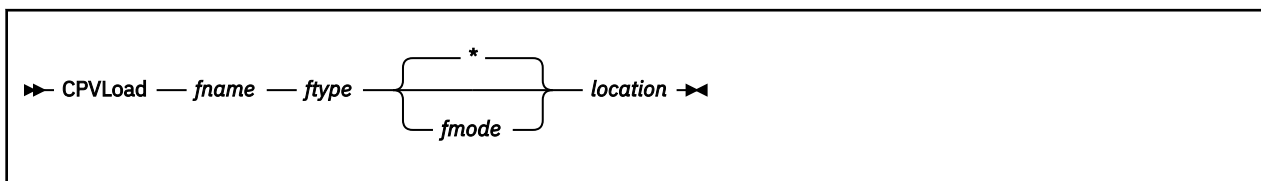
```
xx:
```

is prefixed to each output line to identify the processor whose execution of *command\_line* produced this line.

### **Messages**

- HCP001E Unknown CP command: *command*
- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP1454E CPU *nn* does not exist.
- HCP1455E CPU *nn* is not started because it is in check-stop state.
- HCP1460E Only one of the CPU commands may specify multiple virtual CPUs when entering nested CPU commands.

## CPVLOAD



### Authorization

Privilege Class: B

### Purpose

Use CPVLOAD to load a file into your virtual machine's storage.

### Operands

#### *fname*

is the name of the file you want loaded.

#### *fmode*

is the type of the file you want loaded.

\*

tells CP to search the list of accessed disks for the file you want to load. This is the default if you do not specify a file mode.

#### *fmode*

is the mode of the CP-accessed minidisk where the file you want to load resides.

#### *location*

is the 31-bit address, in hexadecimal, where the file is to be loaded.

### Usage Notes

1. The specified file is loaded as a core image into virtual machine storage. The contents of the file are copied into virtual storage starting at the designated location and proceeding upward.
2. The file may be fixed or variable format and must have a logical record length less than 4,000 bytes.
3. Before you can use the CPVLOAD command to load a file into virtual storage, someone already must have issued a CPACCESS command or specified a CP\_ACCESS statement to identify the minidisk to CP. Although you do not have to link to that minidisk, your virtual machine must be capable of establishing one without a password. For more information on the CPACCESS command, see [“CPACCESS” on page 89](#). For more information on the CP\_ACCESS Statement, see [z/VM: CP Planning and Administration](#).
4. To display a list of the minidisks that CP has access to, use the QUERY CPDISKS command. For more information, see [“QUERY CPDISKS” on page 824](#).

### Responses

Response 1:

```
X'nnnnnnnn' bytes loaded
```

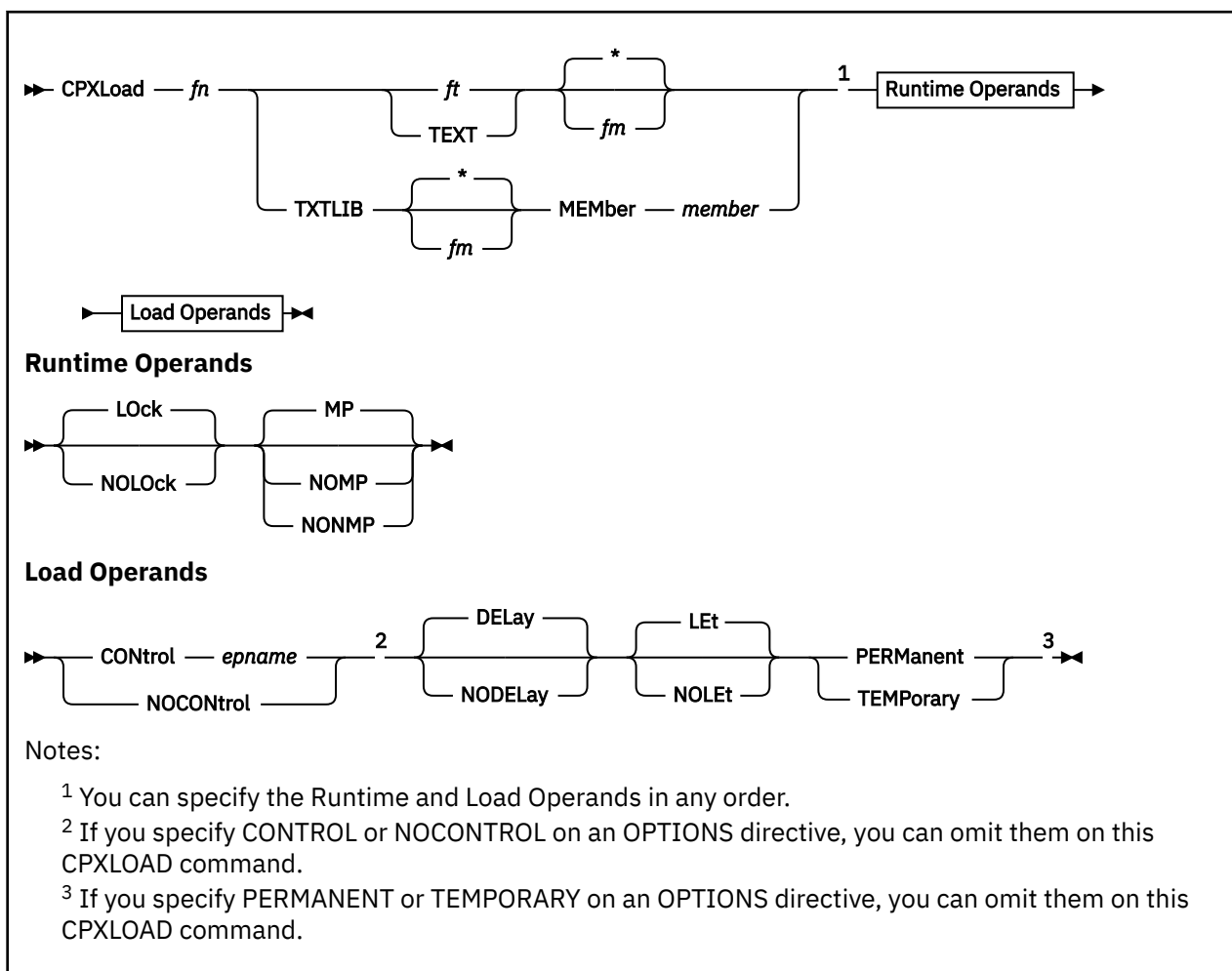
### Messages

- HCP001E Unknown CP command

## CPVLOAD

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP004E Invalid hexloc - *nnn*
- HCP013E Conflicting option - *option*
- HCP410E Command terminated; paging error
- HCP2766E Attempt to open *fname ftype fmode* failed, return code from open is *code*
- HCP6703E File *fname ftype fmode* not found
- HCP6704E Missing token at end of line
- HCP6706E Invalid file mode - *string*
- HCP6706E Invalid file name - *string*
- HCP6706E Invalid file type - *string*
- HCP6706E Invalid record length - *string*
- HCP6706E Invalid string - *string*
- HCP6709E Too many items specified - *item*
- HCP6740E Insufficient storage to complete this request

## CPXLOAD



### Authorization

Privilege Class: A

### Purpose

Use CPXLOAD to load a file containing customer-written CP routines into the system execution space.

You can also load customer-written CP routines into the system execution space during initialization using the CPXLOAD configuration file statement. For more information on the [CPXLOAD Statement](#), see *z/VM: CP Planning and Administration*.

### Operands

***fn***

is the name of the file that you want loaded.

***ft***

is the file type (other than TEXT or TXTLIB) of the file that you want loaded.

This file may contain CPXLOAD directives, text records, or a combination of both. If the file contains text records, it must have a fixed record format (RECFM F) and a logical record length of 80 (LRECL

80). If the file does not contain text records, it can be any record format and must have a logical record length less than 4,000 (LRECL < 4000).

**TEXT**

tells CP that this is a text file that contains 1 or more CSECTs and can contain 1 or more CPXLOAD directives. Text files must have a fixed record format (RECFM F) and a logical record length of 80 (LRECL 80).

**TXTLIB**

tells CP that the file is a text library that contains 1 or more members. A TXTLIB member can contain 1 or more control sections (CSECTs) and can contain CPXLOAD directives. TXTLIB files must have a fixed record format (RECFM F) and a logical record length of 80 (LRECL 80).

**\***

tells CP to search the list of CP-accessed minidisks until it finds the first occurrence of the specified file that you want loaded. If you do not specify a file mode, \* is the default.

***fm***

is the file mode of the CP-accessed minidisk containing the file that you want loaded.

**MEMBER *member***

is the name of the member in the TXTLIB that you want loaded.

You can use generic member names to request a specific subset of files. A generic member name is a 1- to 8-character string with asterisks (\*) in place of 1 or more characters and percent signs (%) in place of exactly 1 character. For example:

```
hc%p* . . .
```

lists all members that start with HC and have P as their fourth character.

**LOCK****NOLOCK**

has no effect and is retained only for compatibility. All symbols are considered resident, which means they cannot be locked or unlocked.

**MP**

tells CP that the entry point is multiprocessor (MP) capable. This means that the entry point can be dispatched on any of the machine's processors. If omitted, MP is the default.

**NOMP****NONMP**

tells CP that the entry point is dispatched only on the master processor, because (in general) the entry point assumes that competitive routines are also not multiprocessor (MP) capable. Use NOMP or NONMP to prevent entry points from overlaying each other's chains of control blocks when you do not take the precaution of getting a system lock. For example, SPOOL routines are NOMP.

**CONTROL *epname***

tells CP to call the specified entry point after loading the customer-written CP routines and before processing a CPXUNLOAD request. You can load the customer-written CP routines containing the specified entry point either before or within this CPXLOAD request. The variable *epname* must be a 1- to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

**Note:** Normally, if CP cannot find an entry point when processing an exit point routine, it ignores the unknown entry point and continues normal processing. This is not true when you specify the CONTROL *epname* operand. If CP cannot find the entry point you specify on CONTROL, CP will terminate processing your CPXLOAD command and will not load your customer-written CP routines into the system execution space.

For information about control entry points, see *z/VM: CP Exit Customization*.

**NOCONTROL**

tells CP not to call an entry point after loading the customer-written CP routine and before processing a CPXUNLOAD request.



**DElay**

This operand only has meaning on the CPXLOAD configuration file statement, where it tells CP not to process a CPXLOAD statement until after all CP\_ACCESS statements have been processed. It is included here, on the command, for compatibility with the CPXLOAD statement. If you specify DELAY, CP ignores the operand, issues informational message HCP2777I, and loads the CP routines immediately.

**NODElay**

This operand only has meaning on the CPXLOAD configuration file statement, where it tells CP to process a CPXLOAD statement immediately, without waiting for all CP\_ACCESS statements to be processed. It is included here, on the command, for compatibility with the CPXLOAD statement. If you specify NODELAY, CP ignores the operand, issues informational message HCP2777I, and loads the CP routines immediately.

**LEt**

tells CP to load the specified file and to ignore any records that are completely blank or that contain an unexpected value in column 1. This is meant to accommodate the noncommented information that can be left in a TEXT file by an assembler utility such as VMFASM. If omitted, LET is the default.

**NOLEt**

tells CP to stop loading the specified file when it encounters an unexpected value in column 1. Column 1 is expected to contain '\*' (to denote a comment), X'02' (to denote a TEXT record), or blank (to denote a possible CPXLOAD directive).

**PERManent**

tells CP that the customer-written CP routines being loaded are to remain a part of CP until a CP SHUTDOWN command is issued or a software-initiated restart (bounce) occurs. This means you cannot use the CPXUNLOAD command to remove these CP routines.

**TEMPorary**

tells CP that the customer-written CP routines being loaded can be unloaded in the future with a CPXUNLOAD command.

**Usage Notes**

1. When loading your files into storage, CP treats each control section (CSECT) independently for storage allocation. Also during loading, CP allocates one page of storage to each CSECT. There is one exception: if CP encounters a CSECT of zero length during CPXLOAD processing, that CSECT is deleted. If you need to load a zero-length CSECT, add an EXPAND directive to your input file. For example, if you had zero-length CSECT XXXDOG to load, you would add "EXPAND XXXDOG(8)" to your input file.
2. When invoking your files, CP treats each entry point in a CSECT independently for the MP attribute.
3. The customer-written CP routines that you are loading must be on a disk that CP has access to when the load operation is performed. If you have a CPXLOAD statement in your system configuration file that specifies the NODELAY operand, the customer-written CP routines must be on the parm disk because the minidisks specified on CP\_ACCESS statements are not available until the end of the initialization process. If your CPXLOAD statement specifies the DELAY operand, the customer-written CP routines can be a file on any disk, as long as you have specified a CP\_ACCESS statement for that disk in your system configuration file.
4. You can specify runtime and load operands on either the CPXLOAD command or the OPTIONS directive, or both. However, if you specify options on both and those options conflict, CP uses the options from the CPXLOAD command. For example, suppose you specify PERMANENT on the OPTIONS directive and TEMPORARY on the CPXLOAD command, CP will load the CP routines as temporary. For more information about the [OPTIONS Directive](#) and several other directives, see *z/VM: CP Exit Customization*.
5. To assign entry points and external symbols to an exit point and to enable or disable that exit point, use the ASSOCIATE EXIT command (see ["ASSOCIATE EXIT"](#) on page 20).
6. To assign an external symbol to a local message repository, use the ASSOCIATE MESSAGES or MSGS command (see ["ASSOCIATE MESSAGES / MSGS"](#) on page 24).

7. To display information about customer-written CP routines loaded by the CPXLOAD command, use the QUERY CPXLOAD command (see [“QUERY CPXLOAD”](#) on page 848).
8. To display information about external symbols you may have loaded, use the LOCATE SYMBOL command (see [“LOCATE SYMBOL”](#) on page 636).
9. To remove CPXLOADed files from the system execution space:
  - a. Use the DISASSOCIATE command (see [“DISASSOCIATE”](#) on page 324) to revoke all entry point and external symbol assignments made with the ASSOCIATE EXIT, MESSAGES, or MSGS commands or configuration file statements
  - b. Use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117) to unload the customer-written CP routines.

For more information about the ASSOCIATE EXIT Statement or ASSOCIATE MESSAGES / MSGS Statement, see *z/VM: CP Planning and Administration*.

10. For more information about loading customer-written CP routines into the system execution space, about runtime and load operands, and about CPXLOAD Directives, see *z/VM: CP Exit Customization*.
11. CPXLOAD provides the ability to load executable code, message repositories, and data modules dynamically. Only compiled files may be loaded. These compiled files would be the TEXT file output from the assembler or from the CMS GENMSG command.

## Responses

Response 1:

To permanently load a multiprocessor capable abend text deck, enter the following:

```
cpxload abend text * mp control kaos permanent
Loaded as identifier 7
Ready;
```

Note that CP assigns a load identifier (load ID) to the loaded CP routines. Please make a note of this number. If you ever want to unload these CP routines (using the CPXUNLOAD command), you will need to specify the load ID that CP assigned. If you do not know the load ID, use the ALL operand of the QUERY CPXLOAD command to display (among other things) the load IDs of all the CP routines loaded onto (and not yet unloaded from) the system.

The load ID that CP assigns is a 1- to 10-digit decimal number between 0 and 2,147,483,647. The first time you use the CPXLOAD command (or configuration file statement), CP assigns that set of customer-written CP routines a load ID of 0. CP increases the load ID by 1 for each subsequent CPXLOAD request. If all the CPXLOAD requests are successful, you will have a sequential list of loaded CP routines.

If one or more of the CPXLOAD requests are unsuccessful or you issue the CPXUNLOAD command to unload some customer-written CP routines, there will be one or more gaps in your sequential list of loaded CP routines. The next time you load some customer-written CP routines, CP ignores these gaps in the list and assigns the newly-loaded CP routines a load ID that is one more than the last assigned load ID.

For example, suppose you had customer-written CP routines loaded at IDs 0 through 7 and you unloaded the CP routines at IDs 2, 3, and 5. This means you still have CP routines loaded at IDs 0, 1, 4, 6, and 7. The next time you use the CPXLOAD command, CP will assign 8 as the load ID for those CP routines. CP will not try to fill in the gaps at 2, 3, or 5.

## Messages

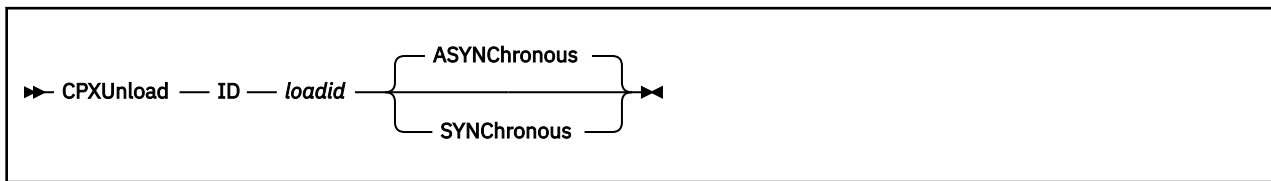
- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP410E Request terminated; paging error

- HCP1014E A required option is missing - *option*
- HCP2753E The specified control entry point caused CPXLOAD to fail  
Cannot call the specified entry point
- HCP2753E The specified control entry point caused CPXLOAD to fail  
It is a TEMPORARY entry point
- HCP2753E The specified control entry point caused CPXLOAD to fail  
The specified entry point rejected the CPXLOAD request with return  
code *nn*
- HCP2753E The specified control entry point caused CPXLOAD to fail  
The specified entry point was marked for deletion
- HCP2753E The specified control entry point caused CPXLOAD to fail  
The call-by-register vector in the HCPENTER macro expansion  
for the specified entry point was not valid
- HCP2755E Encountered I/O error while loading cards
- HCP2756E Encountered verify card failure
- HCP2757E This option {, or its opposite} has already been specified - *option*
- HCP2759E Symbol *symbol* is referenced in this file, but was defined in a TEMPORARY CPXLOAD file
- HCP2760E Symbol *symbol* is defined in this TEMPORARY file, but CPXLOAD ID *id* refers to it
- HCP2761E Encountered duplicate member in file *fn ft fm*
- HCP2766E Attempt to open file *fn ft fm* failed, return code from open is *nnnn*
- HCP2767E Nothing was loaded
- HCP2768E Missing {file name|file type|number|entry point name|file member name}
- HCP2772E Relocatable address constant is not 4 bytes long
- HCP2777I {Operand|Option} {DELAY|NODELAY} has been ignored
- HCP6702E Error encountered while attempting to read records from *fn ft*.
- HCP6703E File *fn ft fm* member *member* not found
- HCP6703E File *fn ft fm* not found.
- HCP6704E Missing token at end of line
- HCP6706E Invalid entry point name - *name*
- HCP6706E Invalid file member name - *member*
- HCP6706E Invalid file mode - *fm*
- HCP6706E Invalid file name - *fn*
- HCP6706E Invalid file type - *ft*
- HCP6710E Including file *fn ft* would cause an INCLUDE loop -- statement ignored
- HCP6740E Insufficient storage to complete this request
- HCP8000E File *fn ft fm* not found
- HCP8001E File *fn ft fm* is not fixed, 80 character records
- HCP8003E Data not replaced: *text*
- HCP8005E Duplicate CSECT - *csect*
- HCP8011E Verify failure:
- HCP8012E Already defined CSECT: *csect*
- HCP8013I Possible overlay:
- HCP8014E Duplicate ENTRY: *text*
- HCP8015E Invalid card format:
- HCP8016E Invalid card:

## CPXLOAD

- HCP8017E Invalid date:
- HCP8018E Invalid format:
- HCP8019E Invalid placement:
- HCP8020E No argument specified:
- HCP8021E Operand missing:
- HCP8022E Undefined ENTRY:
- HCP8023E Unrecognized card
- HCP8025E No CSECTs defined
- HCP8029E Duplicate ESDID definition; missing 'END' card
- HCP8040E File *fn ft fm* Record *nnn*  
Copy\_of\_record
- HCP8042E ESDID value has been incremented beyond 65535 (X'FFFF')

## CPXUNLOAD



### Authorization

Privilege Class: A

### Purpose

Use CPXUNLOAD to remove customer-written CP routines from the system execution space.

### Operands

#### ID *loadid*

is the identifier that CP assigned to these CP routines when you loaded them into the system execution space using the CPXLOAD command. The variable *loadid* must be a decimal number between 0 and 2147483647.

#### ASYNChronous

tells CP that you want to process other commands while this CPXUNLOAD command completes. ASYNCHRONOUS is the default.

#### SYNChronous

tells CP to prevent you from executing other commands until the CPXUNLOAD command completes.

### Usage Notes

1. To load customer-written CP routines into the system execution space, use the CPXLOAD command (see “CPXLOAD” on page 111) or the CPXLOAD configuration file statement. For more information about the [CPXLOAD Statement](#), see *z/VM: CP Planning and Administration*.
2. Before you unload the customer-written CP routines from the system execution space, use the DISASSOCIATE command (see “DISASSOCIATE” on page 324) to revoke all entry point and external symbol assignments made with the ASSOCIATE EXIT, MESSAGES, or MSGS commands or configuration file statements. For more information about the ASSOCIATE EXIT command, see “ASSOCIATE EXIT” on page 20. For more information about the ASSOCIATE MESSAGES or MSGS commands, see “ASSOCIATE MESSAGES / MSGS” on page 24. For more information about the [ASSOCIATE EXIT Statement](#) or [ASSOCIATE MESSAGES / MSGS Statement](#), see *z/VM: CP Planning and Administration*.
3. You can unload customer-written CP routines from the system execution space only if you specified the TEMPORARY operand of the CPXLOAD command or configuration file statement. For more information about the CPXLOAD command, see “CPXLOAD” on page 111. For more information about the [CPXLOAD Statement](#), see *z/VM: CP Planning and Administration*.
4. To unload customer-written CP routines from the system execution space, you must specify the load ID that CP assigned to the CP routines when loading them. CP assigns and displays this load ID in the response to your CPXLOAD command. If you cannot remember the load ID, use the QUERY CPXLOAD ALL command (see “QUERY CPXLOAD” on page 848).
5. For more information about unloading customer-written CP routines from the system execution space, see [Understanding the System Execution Space and CP Customization Methods](#) in *z/VM: CP Exit Customization*.

## Responses

Response 1:

To unload the customer-written CP routines that was assigned ID 92, enter the following:

```
cpxunload id 92
HCP2770I CPXUNLOAD for load ID 92 has been scheduled
Ready;
:
HCP2771I CPXUNLOAD for load ID 92 has been completed
```

Note that the Ready message in this response appears between the two messages. The virtual machine issuing this CPXUNLOAD command did not specify anything after the load ID, so this became an asynchronous command, by default. This means the virtual machine is free to issue other commands before CP unloads all the customer-written CP routines. When you specify SYNCHRONOUS, your virtual machine cannot perform other tasks until CP unloads all of the customer-written CP routines.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP2769E CPXUNLOAD for load ID *loadid* has been rejected; the requested load ID was not found
- HCP2769E CPXUNLOAD for load ID *loadid* has been rejected; the requested load ID was loaded as PERMANENT
- HCP2769E CPXUNLOAD for load ID *loadid* has been rejected; the requested load ID is still in use by a system service
- HCP2769E CPXUNLOAD for load ID *loadid* has been rejected; the request was denied by the control entry point
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*

## DEACTIVE CONV



### Authorization

Privilege Class: B

### Purpose

Use the DEACTIVE CONV command to deactivate a conversation between transaction programs that is being managed by the local VM node.

To deactivate a collection management conversation between nodes, use the DEACTIVE ISLINK command (see [“DEACTIVE ISLINK”](#) on page 120).

There are two types of conversations in the Communication Services (CS) collection:

- Conversations between nodes are used for communications management and CS collection control. These conversations are called collection management conversations (CMP conversations).
- The other type of conversations in the CS collection are conversations between user transaction programs and resource transaction programs. These types of conversations are called APPC conversations (a reference to the program's use of the assembler language or CPI Communications programming interface to APPC/VM).

### Operands

#### *convid*

Is the hexadecimal conversation identifier of a specific conversation managed by the local VM node. To determine the conversation ID, enter the QUERY CONV command (see [“QUERY CONV”](#) on page 810).

### Responses

Response 1:

```
Conversation convid is deactivated.
```

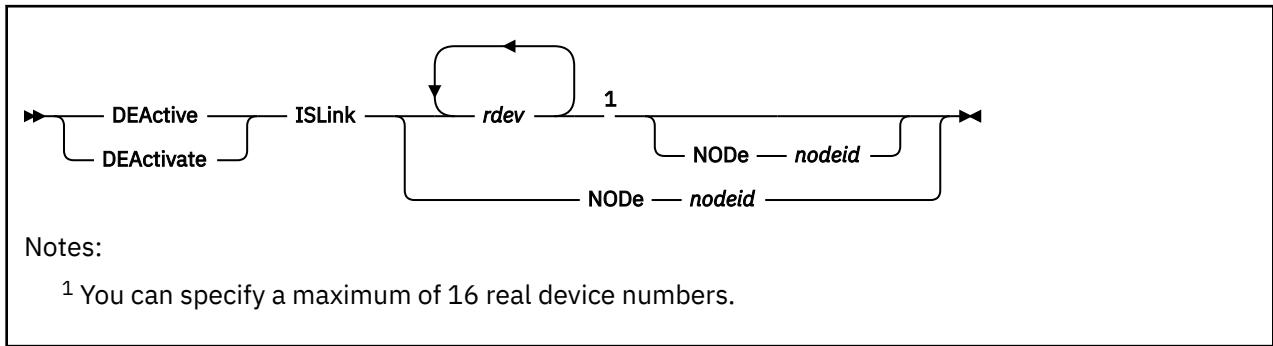
#### *convid*

Identifies the conversation ID of the conversation that is deactivated.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP2706I Conversation *convid* deactivated by user *userid*
- HCP2728E Conversation *convid* does not exist.
- HCP2729E Conversation *convid* is for collection management. Conversation not deactivated.

## DEACTIVE ISLINK



### Authorization

Privilege Class: B

### Purpose

Use the DEACTIVE ISLINK command to deactivate active conversations on one or more devices.

To deactivate a conversation between transaction programs, use the DEACTIVE CONV command.

### Operands

#### *rdev*

Identifies one to sixteen real device numbers to be deactivated.

When specified with NODe *nodeid*, devices are deactivated only if the link configuration matches *nodeid*.

#### **NODe** *nodeid*

Specifying NODe *nodeid* without an *rdev* deactivates all devices.

### Usage Notes

1. The system operator receives all informational and error messages, but not immediate command responses.
2. When you deactivate all remaining devices associated with an ISLINK, that ISFC link to a neighboring node is disabled. The affected link may be your only connection to this neighbor. It might also be the only connection to other nodes in the ISFC collection. Losing access to a node in the ISFC collection may terminate IUCV or APPC communication with those nodes. For example, a local user editing a file in a Shared File System directory residing on a remote node would lose access to that file while this link is disabled. Even if ISFC can access these remote resources through another ISFC link, IUCV and APPC sessions will be terminated by this disruption and the guest will have to take action to regain access to the remote node.
3. Avoid using DEACTIVATE ISLINK to remove a link device from an ISFC logical link while a VMRELOCATE MOVE command is in progress. Removing a link device interferes with the relocation and can cause operations to time out, terminating the relocation.

### Responses

Response 1:

```
Link rdev deactivated.
```



***rdev***

is the link address that has been deactivated.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP2701I Link *linkname* went down.
- HCP2703I Node *nodeid* deleted from collection.
- HCP2706I Link *linkname* deactivated by user *userid*.
- HCP2720E Link device *rdev* not found.
- HCP2720E Node *nodeid* not found.
- HCP2738E Deactivate in progress for link *linkname* -- command rejected.
- HCP2746E Deactivate failed. Node *node1* was specified on the command, but link device *rdev* is connected to node *node2*.

## DEFINE

---

### Purpose

Use the DEFINE commands to:

- Change the configuration of your virtual machine
- Change the configuration of your operating system
- Add a new alias for an existing CP command on your system
- Add a new CP command to your system
- Add a new version of an existing CP command to your system
- Add a new DIAGNOSE code to your system
- Add a new guest LAN to your system.

### Usage Notes

(General for DEFINE)

1. SET IMSG OFF suppresses informational messages for this command.
2. When you use the DEFINE command to alter the configuration of your virtual machine, the changes are temporary and are in effect only for the current session.
3. When you use the DEFINE command to alter the real I/O configuration of the processor, the changes are temporary and are in effect only until the next power-on reset (POR). If you want the changes to be permanent, you must update your I/O configuration program (IOCP) file.
4. Defaults for some real printer attributes (for example CHARS) are not set when a virtual printer is created with the DEFINE command.

### Messages

(General for DEFINE)

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*
- HCP005E Invalid hexdata - *hexdata*
- HCP007E Invalid userid - *userid*
- HCP010E Invalid register - *register*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP023E VOLID missing or invalid
- HCP024E {CYLINDERS|PAGES} MISSING OR INVALID {Block number|Cylinder|Cylinder or block numbers} missing or invalid
- HCP026E Operand missing or invalid
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP045E *userid* not logged on
- HCP052E [XAUTOLOG failed for *userid*:] Error in CP directory
- HCP053E [XAUTOLOG failed for *userid*:] *userid* not in CP directory
- HCP091E DASD *vdev* not defined; *{temp|vdisk}* space not available

- HCP091E DASD *vdev* not defined; requested vdisk exceeds DEFINE VFB-512 maximum permitted blocks: 4194296.
- HCP092E Device *vdev* not defined; device *vdev* already defined
- HCP094E Storage exceeds allowed maximum of *maximum* {M|K}
- HCP101E DASD *vdev* forced R/O; R/O by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[; stable by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP102E DASD *vdev* forced R/O; R/W by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[; stable by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP103E DASD *vdev* forced R/O; R/W by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}, R/O by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[; stable by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP104E *userid* *vdev* not linked; R/O by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[; stable by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}|exclusive by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP105E *userid* *vdev* not linked; R/W by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[; stable by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}|exclusive by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP106E *userid* *vdev* not linked; R/W by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}, R/O by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*} [; stable by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP107E *userid* *vdev* not linked; not in CP directory
- HCP108E *userid* *vdev* not linked; valid *valid* not mounted
- HCP109E *userid* *vdev* not linked; invalid link device
- HCP110E *userid* *vdev* not linked; *type* *vdev* already defined
- HCP116I WARNING: Cylinders *n* through *n* of minidisk *vdev* are the target of FlashCopy relationship *hhhhhhh*.
- HCP117E Userid *userid* not linked; valid *valid* conflict
- HCP153E Device {*vdev*|*range*} exceeds maximum number of virtual devices
- HCP195E *alias\_name* is already a substitute
- HCP259E SIZE parameter missing or invalid
- HCP260E DEFINE command processing cannot complete due to errors
- HCP349E String supplied does not match storage contents
- HCP372E Invalid location for exit; command rejected
- HCP373E Exit already defined at location; command rejected
- HCP844E The device is not defined due to an invalid control unit type.
- HCP864E A reserved partition slot does not exist for CSS *nn*, MIF-ID *mm*
- HCP864E Partition *pname* is already configured in CSS *nn*, MIF-ID *mm*
- HCP864E Partition *pname* not found
- HCP1003E *module* was not found
- HCP1006E Command processing cannot complete because the system does not support the Dynamic-I/O-for-I/O-adapters facility.
- HCP1006E Command processing cannot complete because the system does not support the coupling over InfiniBand channel-path facility.
- HCP1006E Command processing cannot complete because the system does not support the OSA-NCP-(OSN)-channel-path facility

## DEFINE

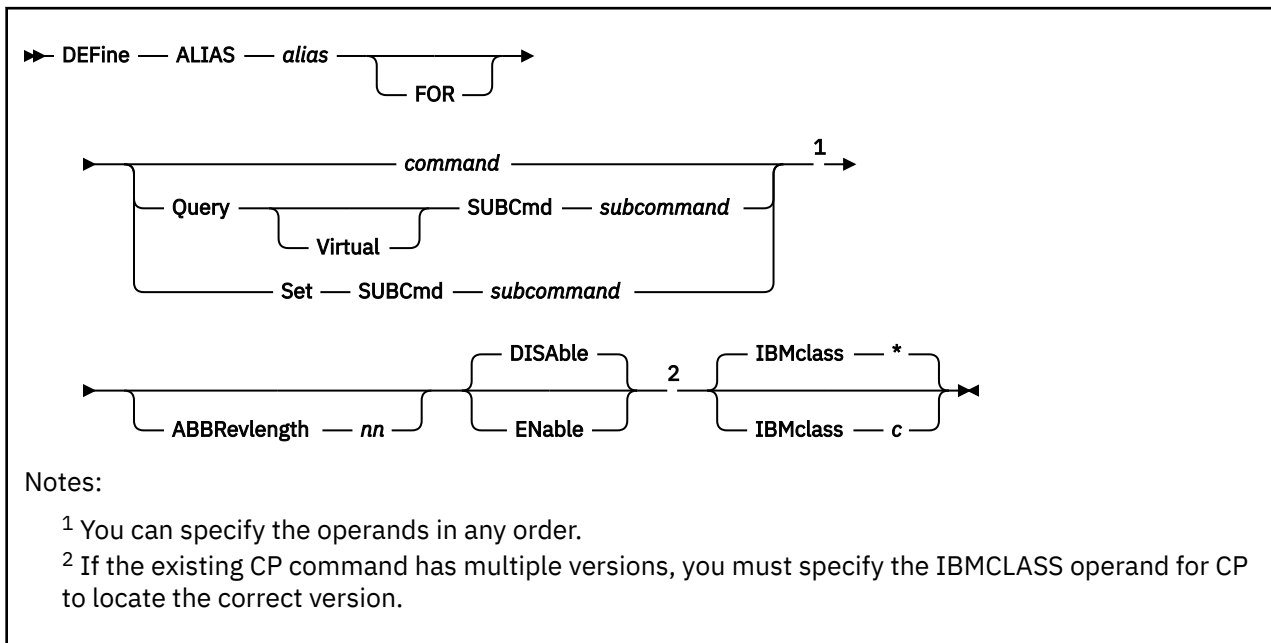
- HCP1006E The command processing cannot complete because the system does not support the logical-partition-configuration facility
- HCP1006E The command processing cannot complete because the system is not running in EMIF mode
- HCP1006E Command processing cannot complete because the system does not support the OSC 3215-type device facility.
- HCP1006E Command processing cannot complete because the system does not support the OSX and OSM Channel-Path-Types facility.
- HCP1006E Command processing cannot complete because the system does not support the Utility-String facility.
- HCP1006E Command processing cannot complete because the system does not support the multiple-subchannel-set facility.
- HCP1014E A required option is missing.
- HCP1015E [XAUTOLOG failed for *userid*:] Insufficient storage is available to satisfy your request.
- HCP1150E DASD {*rdev/vdev*} is not a valid base exposure.
- HCP1151E *userid vdev* has not been linked because it is not within CP volume cylinder extents.
- HCP1152E *userid vdev* has not been linked because it would overlap system {directory|spool|paging|temporary disk} space
- HCP1200E DASD *vdev* has not been redefined because paging subsystem device cannot be redefined.
- HCP1453E CPU *nn* is not defined because it already exists.
- HCP1454E CPU *nn* does not exist.
- HCP1458E An attempt was made to define more CPU's than is allowed in your directory.
- HCP1462E An attempt was made to define CPU(s) that would create a virtual CPU configuration that is not valid.
- HCP1464I CPU affinity is suppressed for CPU *n*
- HCP1466I CPU affinity is no longer suppressed for CPU *n*
- HCP1467E Virtual *type* processors are not supported on your system.
- HCP2550I The console has been defined as a 3215 because a console mode of 3270 is not supported by the VTAM® Service Machine controlling the real terminal.
- HCP2551E The console has been defined as a 3215 because the real device is not a 3270 display.
- HCP2773I Entry point name *name* is unknown at this time; processing continues
- HCP2774E Multiple commands match this [alias] request (*command1, command2*)
- HCP2775E External security virtual machine is running. You cannot define new commands at this time
- HCP2781E *vdev* not created; network devices *vdev-lastdev* could not be defined
- HCP2792E OSA *vdevnum* has not been redefined because it is part of a virtual network adapter
- HCP2800E *userid* is not an authorized CF Service Machine.
- HCP2801E Unable to define 4 consecutive message devices starting with device *nnnn*.
- HCP2802E The maximum number of connections to/from *userid* are already defined.
- HCP2803E A coupled connection with CF Service Machine *userid* already exists.
- HCP2804I Message devices *range* defined and coupled to *msgprocid*.
- HCP2806E You are not allowed to define any message processors
- HCP2806E You are not allowed to define message processor *msgprocid*
- HCP2811E Message processor *msgprocid* is not operational.
- HCP2823E Number of devices must be even.
- HCP2880E *userid vaddr* is not linked, CSE LINK I/O error, code *cc*.
- HCP6005E Option *option* is not supported by z/VM.

- HCP6296E Devices cannot be defined in subchannel set *m* because subchannel sets 0 and *n* are currently in use.
- HCP6430E DEFINE HYPERPAVALIAS cannot complete because there is at least one base device in this HyperPAV pool that is quiesced for Hyperswap.
- HCP6430E DEFINE PAVALIAS cannot complete because the associated PAV base is quiesced for Hyperswap.
- HCP6704E Missing token at end of line
- HCP6706E DASD *vdev1-vdev2* defined
- HCP6706E Invalid address range - *range*
- HCP6706E Invalid alias name - *alias*
- HCP6706E Invalid command name - *command*
- HCP6706E Invalid CP Exit number - *exit*
- HCP6706E Invalid Diagnose code - *diag*
- HCP6706E Invalid entry point name - *name*
- HCP6706E Invalid file name - *fn*
- HCP6706E Invalid IBM class - *c*
- HCP6706E Invalid instruction - *instruction*
- HCP6706E Invalid number - *nnnn*
- HCP6706E Invalid privilege classes - *classes*
- HCP6706E Invalid routine name - *routine*
- HCP6706E Invalid subcommand name - *subcommand*
- HCP6706E Invalid time zone ID - *zoneid*
- HCP6706E Invalid time zone offset - *offset*
- HCP6706E Invalid volume label - *valid*
- HCP6779E Volume *valid* already defined in slot *nnn*
- HCP6780E Slot number cannot exceed the defined maximum of *mmm*
- HCP6781E Requested slot occupied by mounted volume *valid*
- HCP6789E Diagnose code *diag* is already defined
- HCP6789E Requested CP Exit number *exit* is already defined
- HCP6805E Cannot determine partition being changed by this dynamic I/O request
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request [for channel path | for control unit | for device | for logical partition] failed with return code *nnnn*
- HCP6806E Dynamic I/O change request for PCI function failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are being controlled by HCD
- HCP6816E Dynamic I/O changes are not allowed because LPAR information is not available
- HCP6816E Dynamic I/O changes are not allowed because the configuration token is not a valid VM token
- HCP6816E Dynamic I/O changes are not allowed for CSS *nn*
- HCP6816E Dynamic I/O changes are not allowed for device *rdev*
- HCP6816E Dynamic I/O changes are not allowed on this system
- HCP6816E Hot I/O rates are being controlled by HCD
- HCP6817E You must specify an existing CP command for this alias

## DEFINE

- HCP6819E A dynamic I/O command is currently executing for user *userid*
- HCP6830I *valid* is currently attached to SYSTEM. Detach the device from SYSTEM and try this command again.
- HCP6834E Command parameter list length is too {small|large}.
- HCP6835E Offset to alias name exceeds parameter list length.
- HCP6835E Offset to alias name is negative.
- HCP6835E Offset to alias name overlays parameter list fields.
- HCP6835E Offset to command name exceeds parameter list length.
- HCP6835E Offset to command name is negative.
- HCP6835E Offset to command name is zero.
- HCP6835E Offset to command name overlays parameter list fields.
- HCP6835E Offset to subcommand name exceeds parameter list length.
- HCP6835E Offset to subcommand name is negative.
- HCP6835E Offset to subcommand name overlays parameter list fields.
- HCP6836E {Command|Subcommand|Alias} name extends beyond end of parameter list.
- HCP6838E {Command|Subcommand} name abbreviation is too large.
- HCP6839E New command or abbreviation duplicates an existing one (*command*)
- HCP6840E You specified QUERY VIRTUAL without SUBCMD
- HCP6840E You specified SUBCMD without SET or QUERY
- HCP6841E You specified IBM class H, which is reserved for IBM use only.
- HCP6841E You specified more than one IBM class.
- HCP6842E Cannot find existing CP command for new alias
- HCP6843E All versions of the command must have the same logon value
- HCP6844E All versions of the command must have the same {PROC value|entry point name|entry point address|minimum abbreviation|SILENT setting}
- HCP6845E You must specify {an entry point name|a privilege class[other than 0]|an IBM class [other than 0]}
- HCP6846E Requested entry point name is not {an external label in the nucleus|at the address indicated}
- HCP6847E You may not define a command whose alias is itself an alias
- HCP6853E CHPID *chpid* has been created but is in definition error and will be unusable when varied on
- HCP6854E CU *cu* has been created but is in definition error. Devices on the CU are not operational and cannot be varied online.
- HCP6867E An internal processing error occurred.
- HCP6867E Device *vdev* is not a base HyperParallel Access Volume.
- HCP6867E Device *vdev* is not a base Parallel Access Volume.
- HCP6867E Device *vdev* is not a minidisk.
- HCP6867E There are no available system attached real alias HyperParallel Access Volumes for device *vdev* to be associated.
- HCP6867E There are no available system attached real alias Parallel Access Volumes for device *vdev* to be associated.
- HCP6873E PCI function *rpfid* has been created but is in definition error and has not been added to the configuration.
- HCP9300E A system soft abend *code* has occurred on CPU *nnnn* for user *userid*.

## DEFINE ALIAS



### Authorization

Privilege Class: A

### Purpose

Use DEFINE ALIAS to define a new alias for an existing CP command on the system.

You can also define a new alias during initialization using the DEFINE ALIAS configuration file statement. For more information, see [DEFINE ALIAS Statement in z/VM: CP Planning and Administration](#).

### Operands

#### *alias*

is the name of the alias that you are defining. The variable *alias* must be a 1- to 12-character string.

#### *command*

is the name of the existing CP command for which you are creating an alias. The variable *command* is a 1- to 12-character string.

#### **Query SUBCmd *subcommand***

tells CP the name of the existing CP QUERY subcommand for which you are creating an alias. The variable *subcommand* is a 1- to 12-character string.

#### **Query Virtual SUBCmd *subcommand***

tells CP the name of the existing CP QUERY VIRTUAL subcommand for which you are creating an alias. The variable *subcommand* is a 1- to 12-character string.

#### **Set SUBCmd *subcommand***

tells CP the name of the existing CP SET subcommand for which you are creating an alias. The variable *subcommand* is a 1- to 12-character string.

#### **ABBRlength *nn***

is the length of the smallest acceptable abbreviation of the alias that you are defining. The variable *nn* is a decimal number between 1 and the length of the full alias name.

### **DISable**

tells CP not to activate this alias until you enable it. (For more information about enabling aliases, see Usage Note “5” on page 129.) If omitted, DISABLE is the default.

### **ENable**

tells CP to immediately activate this alias.

### **IBMclass \***

tells CP to define aliases for all versions of the specified command or subcommand. If omitted, IBMCLASS \* is the default.

### **IBMclass c**

tells CP to define an alias for a specific version of the specified command or subcommand. The variable *c* can be any 1 of the following:

#### **A**

this is a system-control command to be used by the primary system operator.

#### **B**

this is a command for operational control of real devices.

#### **C**

this is a command to alter system storage.

#### **D**

this is a command for system-wide control of spool files.

#### **E**

this is a command to examine system storage.

#### **F**

this is a command for service control of real devices.

#### **G**

this is a general-use command used to control the functions of a virtual machine.

#### **O**

(zero) this command has no specific IBM class assigned.

## Usage Notes

1. You can create many aliases for one command, but you cannot create an alias for an alias. For example, the CP MSG command is actually an alias for the CP MESSAGE command:

```
MSG ----> MESSAGE
```

You can create another alias for the MESSAGE command:

```
MSG ----> MESSAGE <---- TELL
```

But you cannot create an alias for MSG:

```
TELL ----> MSG ----> MESSAGE
```

2. If you specify the QUERY, QUERY VIRTUAL, or SET operands, you are creating an alias for a subcommand, not a command. For example, suppose you created alias TUBE for the CP QUERY VIRTUAL GRAF command. To invoke your new alias, you would enter QUERY VIRTUAL TUBE, not QUERY TUBE or just TUBE.
3. When specifying an alias name, you can use special characters in the name. However, we do not recommend that you use the pattern matching characters (\* and %). You can use these characters to define aliases, but they may seem confusing when you issue other commands that allow you to use the pattern matching characters because CP will interpret the % or \* in your alias name as a pattern matching character.

CP places no restriction on the use of special characters such as \*, %, comma, (, ), blank, /, :, \$, #, and @ in command names. However, some External Security Managers (EMs) restrict their use. Please consult your ESM documentation for information on restricted characters.



4. If you try to define a minimum abbreviation that matches the abbreviation for an existing command, subcommand, or alias CP rejects your DEFINE command. For example, if you created a "QUEUE" alias with a minimum abbreviation of 2, CP would reject your QUEUE alias because "QU" is an abbreviation for the QUERY command. In this case, you would need to specify a minimum abbreviation greater than or equal to 4 because the first 3 characters of QUERY and QUEUE are identical.
5. If you do not specify the ENABLE operand, the new alias is initially in a disabled state. To activate an alias after defining it, use the ENABLE COMMAND or CMD command (see [“ENABLE COMMAND / CMD”](#) on page 460).

Note that both the alias you define and its base command must be enabled for the alias to work. For example, if you define alias SHOWIT for the existing command DISPLAY, DISPLAY is considered the base command. DISPLAY must also be enabled for the alias SHOWIT to work.

6. To deactivate an alias after defining it, use the DISABLE COMMAND or CMD command (see [“DISABLE COMMAND / CMD”](#) on page 313).

If you specify the name of an alias on DISABLE COMMAND, then just the alias is disabled. If you specify the name of the base command on DISABLE COMMAND then you disable the entire function. That means that the base command is in a disabled state and every alias you have defined for that command is treated as if it were also disabled.

7. To obtain information such as the enable/disable status and the name of the base command for the alias you have defined, use the QUERY CPCMDS command (see [“QUERY CPCMDS”](#) on page 817).
8. Once defined, an alias name cannot be used again. An alias cannot be modified or eliminated; it can only be disabled. Only a SHUTDOWN or RESTART IPL will eliminate an alias.
9. For more information about creating an alias, see [Defining an Alias Command](#) in *z/VM: CP Exit Customization*.

## Responses

Response 1:

To define GOODNIGHT as an alias for the CP SHUTDOWN command and make it available immediately, enter the following:

```
def alias goodnight for shutdown abbrevlength 5 enable
Ready;
```

Response 2:

To define TELL as an alias for the IBM class <ANY> version of the CP MESSAGE command and make it available immediately, enter the following:

```
define alias tell for message ibmclass 0 enable
Ready;
```

Response 3:

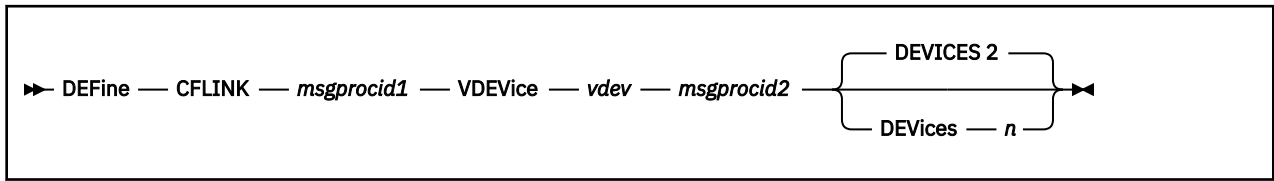
To define NUKE as an alias for all of the IBM class versions of the CP PURGE command and make it available immediately, enter the following:

```
define alias nuke for purge ibmclass * enable
Ready;
```

## Messages

For the list of general messages for the DEFINE command, see [“Messages”](#) on page 122.

## DEFINE CFLINK



### Authorization

Privilege Class: A, B, G

### Purpose

DEFINE CFLINK allows a coupled guest or system operator to create links between coupling facilities.

### Operands

#### *msgprocid1*

specifies the sending coupling facility user ID for the new link.

#### *vdev*

specifies the device number of the first message device in the sending coupling facility. This must be a hexadecimal number from X'0000' to X'FFFE'.

#### *msgprocid2*

specifies the receiving coupling facility for the new link.

#### DEVICES *n*

specifies the number of message devices that the sending virtual machine (coupling guest or coupling facility) can process; *n* must be between 2 and 8. The default is 2.

### Usage Notes

1. A class G only user must be coupled to both the sending and receiving coupling facilities. There also must either be no SPECIAL MSGP statements in the *msgprocid1* directory entry, or *msgprocid2* must appear on one of the SPECIAL MSGP statements in the *msgprocid1* directory entry.
2. A class A or class B user can couple any two coupling facilities regardless of whether they are coupled to the coupling facilities in question and regardless of the presence of SPECIAL MSGP statements in the sending coupling facility's directory entry.
3. A coupling facility can have links to up to seven other coupling facilities.
4. There can be up to two links (channel path IDs) from *msgprocid1* (the sender) to *msgprocid2* (the receiver). In the sender, the first link uses the next available channel path ID in range (E0..E6) and the second uses the next available channel path ID in range (A0..A6). In the receiver, the first link uses the next available channel path ID in range (F0..F6) and the second uses the next available channel path ID in range (B0..B6).
5. The *vdev* chosen must be a hexadecimal number low enough for the addition of *vdev* plus the number of DEVICES minus 1 with a result less than or equal to X'FFFF'.
6. Device numbers X'*vdev*' and X'*vdev*' plus the number of DEVICES minus 1 must be available in the sending coupling facility.
7. If DEVICES is specified without *n*, message HCP026E is displayed.
8. If DEVICES is specified with a device count that is not within the 2 to 8 range, message HCP002E is displayed.
9. If *msgprocid2* is not in z/Architecture mode, CP forces DEVICES to 2 no matter what else was specified.

## Examples

Create a link between CFVM1 and CFVM2, with two message devices in CFVM1 at device numbers 400–401:

```
define cflink cfvm1 vdev 400 cfvm2
```

If CFUSER1 is coupled to both CFVM1 and CFVM2, then CFUSER1 can create a second link between CFVM1 and CFVM2 as follows:

```
define cflink cfvm1 vdev 800 cfvm2
```

**Note:** This creates two more message devices in CFVM1 at addresses 800 and 801.

If CFVM2 was an z/Architecture mode machine, the operator or CFUSER1 could request that more than two devices (up to 8 are allowed) be put in CFVM1 as follows:

```
define cflink cfvm1 vdev 400 cfvm2 devices 6
```

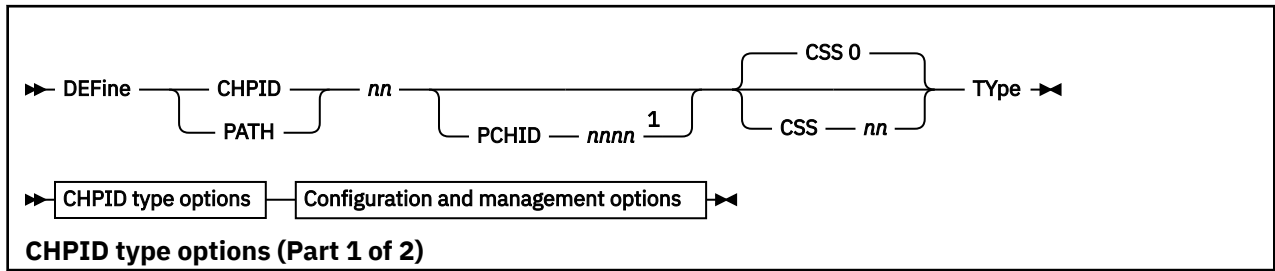
## Messages

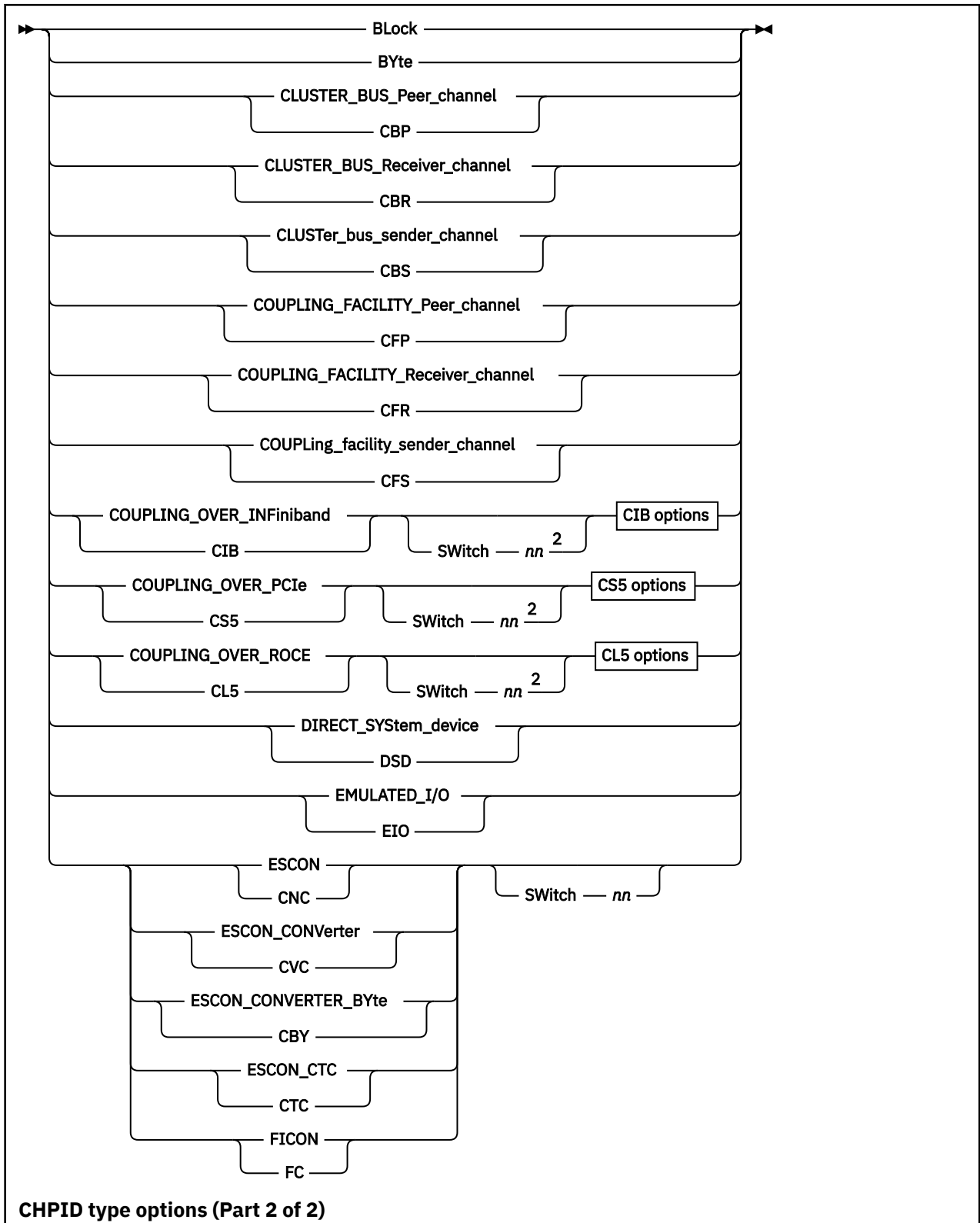
For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

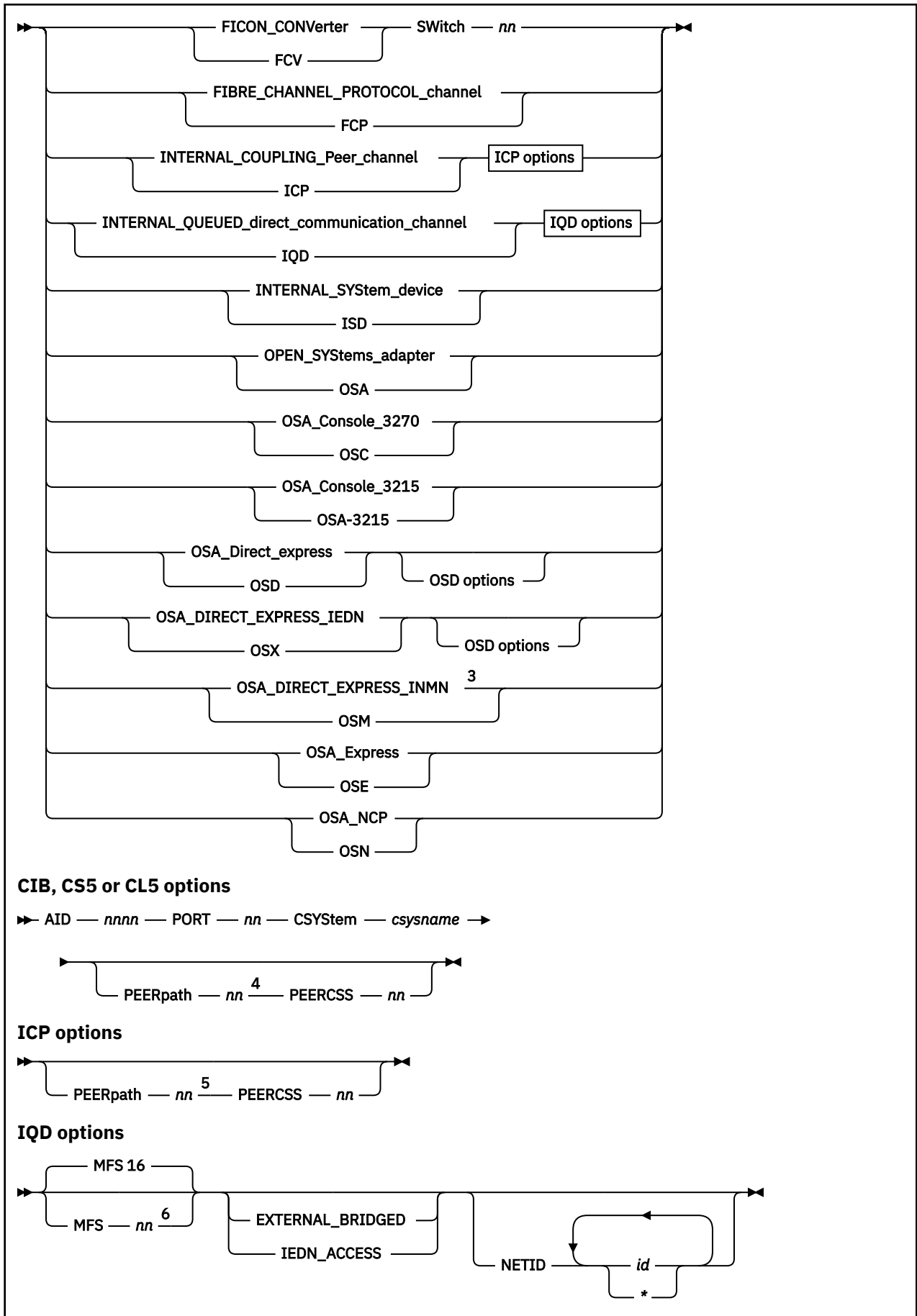
Messages:

- HCP002E Invalid operand — *operand*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP260E Command processing cannot complete due to errors.
- HCP1014E A required option is missing.
- HCP2800E One or more targets is not a CF Service Machine.
- HCP2801E Unable to define consecutive message devices starting with device *nnnn*
- HCP2802E The maximum number of connections {to|from} *userid* are already defined.
- HCP2805E One or more message processors does not exist in your I/O configuration.
- HCP2806E You are not allowed to control message processors.
- HCP2806E The CP directory prohibits this CFLINK.
- HCP2819E CFLINK must not start and end at same CF service machine.
- HCP5769E CFLINK already exists

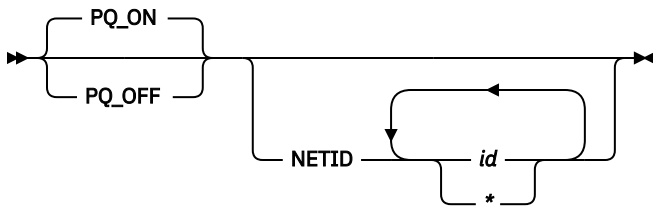
## DEFINE CHPID / PATH



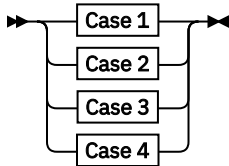




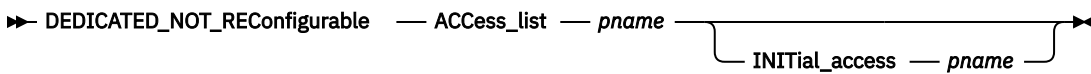
**OSD options**



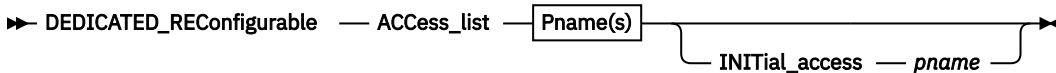
**Configuration and management options**



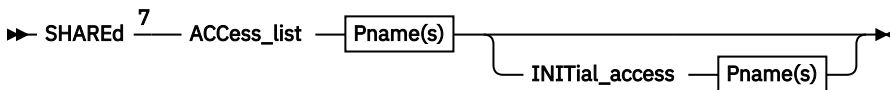
**Case 1**



**Case 2**



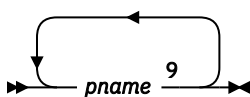
**Case 3**



**Case 4**



**Pname(s)**



**Notes:**

- 1 This operand is required for some CHPID types. See explanation below.
- 2 This parameter is ignored for point-to-point links.
- 3 Priority queuing is always disabled for the Open Systems Adapter Direct Express INMN (OSM) channel paths.
- 4 You must specify this parameter when *csysname* specifies a name other than this system's name or when defining the second path of a pair of connected CIB, CS5 or CL5 paths.
- 5 You must specify this parameter when defining the second path of a pair of connected ICP paths.
- 6 The maximum frame size (MFS) for an IQD path must be a value of 16, 24, 40 or 64 kilobytes.
- 7 This operand is valid only for CBP, CBS, CFP, CFS, CIB, CNC, CS5, CL5, CTC, DSD, EIO, FC, FCP, FCV, ICP, IQD, ISD, OSA, OSC, OSD, OSE, OSM, OSN, and OSX channel paths.
- 8 This operand is valid only for CNC, FC, or FCV switched channel paths.
- 9 The maximum number of logical partition names you can specify depends on the capabilities of your machine.

**Authorization**

Privilege Class: B

## Purpose

Use DEFINE CHPID or PATH to define a new channel path identifier (CHPID) to the I/O configuration.

## Operands

### *nn*

is the number representing the logical channel path (CHPID) that you are defining. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

### **PCHID *nnnn***

is the physical channel ID associated with the channel you are defining. The variable *nnnn* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFF'. Your machine likely supports a lower maximum value. Only PCHIDs that are physically installed on your machine should be entered.

**Note:** This parameter is valid and required only when your machine supports multiple logical channel subsystems. However, this parameter does not apply to InfiniBand host communication adapters (HCAs) or to internal channel-path type of INTERNAL\_COUPLING\_PEER\_CHANNELS (ICP) because there are no physical channel paths associated with them. Unlike ICP channels, this parameter is required when INTERNAL\_QUEUED\_DIRECT\_COMMUNICATION\_CHANNELS (IQD) CHPIDs are being defined to specify a virtual channel identifier on machines that support them.

### **CSS *nn***

is the channel-subsystem ID where the channel path is to be defined. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'. Your machine might support a lower maximum value. See Usage Note [“10” on page 141](#).

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require that the logical channel-subsystem facility be installed.

### **TYPe**

introduces the channel type, defined by the immediately following parameter.

### **BLock**

tells CP that this is a parallel multiplexor channel that interleaves blocks of data.

### **BYte**

tells CP that this is a parallel multiplexor channel that interleaves bytes of data.

### **CLUSTER\_BUS\_Peer\_channel**

#### **CBP**

tells CP that this is a cluster bus peer channel path.

### **CLUSTER\_BUS\_Receiver\_channel**

#### **CBR**

tells CP that this is a cluster bus receiver channel path.

### **CLUSTER\_bus\_sender\_channel**

#### **CBS**

tells CP that this is a cluster bus sender channel path that connects to a cluster bus receiver channel path.

### **COUPLING\_FACILITY\_Peer\_channel**

#### **CFP**

tells CP that this is a coupling facility peer channel path.

### **COUPLING\_FACILITY\_Receiver\_channel**

#### **CFR**

tells CP that this is a coupling facility receiver channel path.

### **COUPLing\_facility\_sender\_channel**

#### **CFS**

tells CP that this is a coupling facility sender channel path that connects to a coupling facility receiver channel path.



**COUPLING\_OVER\_INFiniband****CIB**

tells CP that this is a coupling over InfiniBand channel path.

**COUPLING\_OVER\_PCIE****CS5**

tells CP that this is a coupling over PCIe channel path.

**COUPLING\_OVER\_ROCE****CL5**

tells CP that this is a coupling over RoCE channel path.

**AID *nnnn***

is the adapter identifier of an InfiniBand host communication adapter (HCA) through which a CIB, CS5 or CL5 channel path is defined. The variable *nnnn* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFF'. Your machine might support a lower maximum value. Only adapter identifiers for adapters that are physically installed on your machine should be entered.

**PORT *nn***

is the port number on an InfiniBand host communication adapter (HCA) where a CIB, CS5 or CL5 channel path is defined. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'. Your adapter might support a lower maximum value. Only port numbers for ports that are physically installed on your adapter should be entered.

**CSYSstem *csysname***

is the name of the target system to which a CIB, CS5 or CL5 CHPID connects. It is the name you would see after issuing the QUERY LSYSTEM command in a z/VM system running on that target machine. The name is established via an IOCP configuration definition, HCD configuration definition (in z/OS or z/VM), or the z/VM SET LSYSTEM operator command on the target machine. The variable *csysname* must be a 1- to 8-character string.

**DIRECT\_SYStem\_device****DSD**

tells CP that this is a channel path to a direct system device.

**EMULATED\_I/O****EIO**

tells CP that this is an emulated I/O channel path.

**ESCON****CNC**

tells CP that this channel uses the Enterprise Systems Connection (ESCON) channel to control unit I/O interface to transmit data through optical cables.

**ESCON\_CONVerter****CVC**

tells CP that this is an ESCON channel that attaches to an ESCON converter in block mode.

**ESCON\_CONVERTER\_BYte****CBY**

tells CP that this is an ESCON channel that attaches to an ESCON converter in byte mode.

**ESCON\_CTC****CTC**

tells CP that this ESCON channel path provides channel-to-channel adapter communications.

**FICON****FC**

tells CP that this channel uses the Fibre Connection (FICON) channel to control unit interface to transmit data through optical cables.

**FICON\_CONVerter****FCV**

tells CP that this is a FICON channel that attaches to an ESCON Director with a FICON Bridge card.

**SWitch *nn***

is a number identifying the ESCON or FICON switch to which the channel paths you are defining are connected. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**FIBRE\_CHANNEL\_PROTOCOL\_channel****FCP**

tells CP that this is a SCSI fibre channel protocol (FCP) path.

**INTERNAL\_COUPLING\_Peer\_channel****ICP**

tells CP that this is an internal coupling peer channel path.

**PEERpath *nn***

is the "peer channel path", the number representing the channel-path identifier (CHPID) as known in another machine, or central processing complex (CPC), to which the channel path being defined is connected. The second-to-be-defined path of a pair of connected ICP or CIB paths must specify this parameter. This parameter must also be specified for a CIB path whenever the *csysname* parameter specifies a CPC name other than the name of the system where this command is being executed. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**Note:** If you mistakenly define two connected ICP or CIB paths without defining their association with this parameter, you must delete one of the paths and then redefine it using this parameter.

**PEERCSS *nn***

identifies the channel subsystem where the "peer channel path" is defined. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'. Your machine might support a lower maximum value.

**INTERNAL\_QUEUED\_direct\_communication\_channel****IQD**

tells CP that this is an internal-queued-direct-communication channel. When this channel type is selected, the PCHID operand is also required to be specified to indicate a virtual channel ID on machines that support them.

**MFS *nn***

is the maximum frame size (in kilobytes) to be used by the internal-queued-direct-communication channel being defined. If an MFS value is not specified, the default value of 16 will be used. If an MFS value is specified, the variable *nn* must be a 2-digit decimal number with a value of 16, 24, 40, or 64.

**EXTERNAL\_BRIDGED**

identifies that the specified CHPID can be bridged to a z/VM virtual switch.

**IEDN\_ACCESS**

identifies a single IQD CHPID as a CHPID that is managed by the Unified Resource Manager. When IEDN\_ACCESS is specified, EXTERNAL\_BRIDGED is assumed.

**INTERNAL\_SYSTEM\_device****ISD**

tells CP that this is a channel path to an internal system device.

**OPEN\_SYSTEMS\_adapter****OSA**

tells CP that this is an Open Systems Adapter (OSA) channel path.

**Note:** When defining OSA channels, you should remain in configuration mode while defining the channel path, control unit, and device. If you exit configuration mode before issuing these three DEFINE commands, CP will not add this OSA device to your I/O configuration. For details see the *Planning for the System/390 Open Systems Adapter Feature* book, GC23-3870.

**OSA\_Console\_3270****OSC**

tells CP that this is an OSA Integrated Console Controller (OSA-ICC) channel path (also known as an OSA 3270 console channel path).

**OSA\_Console\_3215****OSA-3215**

dynamically defines an OSC (OSA 3270 console) CHPID for use as a 3215 console.

**OSA\_Direct\_express****OSD**

tells CP that this is an Open Systems Adapter Direct Express (OSD) channel path.

**OSA\_DIRECT\_EXPRESS\_IEDN****OSX**

tells CP that this is an Open Systems Adapter Direct Express IEDN (OSX) channel path.

**PQ\_ON****PQ\_OFF**

tells CP whether or not outbound priority queuing as supported by the QDIO interface is to be used. If PQ\_OFF is specified, then priority queuing is disabled. By not exploiting this function for an OSD channel, the channel can support four (4) times as many queues. For details, refer to Usage Note “11” on page 141. When the option is unspecified, the default is PQ\_ON, which enables priority queuing.

**NETID id****NETID \***

is the network identifier associated with a port of the OSD or IQD channel. A maximum of four network ID's can be specified. The order of the network IDs corresponds to the order of the ports of the channel. An asterisk '\*' can be used as a placeholder that indicates that there is not a network ID associated with a particular port. Each *id* must be a 1- to 16-digit character string.

**OSA\_DIRECT\_EXPRESS\_INMN****OSM**

tells CP that this is an Open Systems Adapter Direct Express INMN (OSM) channel path.

**OSA\_Express****OSE**

tells CP that this is an Open Systems Adapter Express (OSE) channel path.

**OSA\_NCP****OSN**

tells CP that this is an Open Systems Adapter NCP (OSN) channel path.

**DEDICATED\_NOT\_REConfigurable**

tells CP that this is an unshared channel path which can only be configured or deconfigured to the logical partition specified on the ACCESS\_LIST operand.

**DEDICATED\_REConfigurable**

tells CP that this is an unshared channel path which can only be configured to one of the logical partitions specified on the ACCESS\_LIST operand at a time. Later, you can use the MODIFY CHPID or PATH command to reconfigure this channel path to one of the other logical partitions specified on the ACCESS\_LIST operand.

**SHARED**

tells CP that this is a shared channel path which can be concurrently configured to any or all of the logical partitions specified on the ACCESS\_LIST operand. Only the partitions specified in the access list will have access to the defined CHPID. Multiple logical partitions can have initial access to the channel path at device initialization.

**Note:** This operand is valid only for CBP, CBS, CFP, CFS, CIB, CL5, CNC, CS5, CTC, DSD, EIO, FC, FCP, FCV, ICP, IQD, ISD, OSA, OSC, OSD, OSE, OSM, OSN, and OSX channel paths.

**ACCess\_list pname**

specifies the name(s) of the logical partition(s) in which the channel path can be configured in the specified or implied channel-subsystem image. The maximum number of logical partition names you can specify depends on the capabilities of your machine. Each *pname* must be a 1- to 8-digit character alphanumeric string. See Usage Note “1” on page 140 for more information on the maximum number of logical partitions you can specify.

**INITIAL\_access pname**

specifies the name(s) of the logical partition(s) that will have initial access to this channel path in the specified or implied channel-subsystem image. Each pname must be a 1- to 8-digit character alphanumeric string. See Usage Note “1” on page 140 for more information on the maximum number of logical partitions you can specify.

**MANaged**

indicates that the channel path being defined is to be managed by one or more z/OS systems in other logical partitions comprising an I/O cluster. A managed channel path is shared among all logical partitions having access to it, but none have initial access. z/OS dynamically reassigns the channel path within the I/O cluster according to the workload demands within the cluster.

**Note:** This operand is valid only for CNC, FC, or FCV switched channel paths.

**IOcluster iocname**

is the name of the I/O cluster to which the managed channel path is assigned. The variable *iocname* must be a 1- to 8-character alphanumeric string.

**Note:** This operand is valid only for EMIF-capable machines.

**Usage Notes**

1. Specifying Logical Partition Numbers:

- a. When you specify DEDICATED\_NOT\_RECONFIGURABLE, DEDICATED\_RECONFIGURABLE, or SHARED, you must specify at least one logical partition name for ACCESS\_LIST, and you can optionally choose to specify logical partition names for INITIAL\_ACCESS. How many logical partition names you can specify depends upon which operand you choose and the capabilities of your machine; see Table 12 on page 140.

<i>Table 12. Maximum Number of Logical Partition Names</i>		
<b>Operand</b>	<b>ACCESS_LIST</b>	<b>INITIAL_ACCESS</b>
DEDICATED_NOT_RECONFIGURABLE	1	1
DEDICATED_RECONFIGURABLE	MAX	1
SHARED	MAX	MAX

- b. The number of logical partitions that you specify for ACCESS\_LIST must always be greater than or equal to the number of logical partitions you specify for INITIAL\_ACCESS, because the INITIAL\_ACCESS logical partition names are a subset of the ACCESS\_LIST logical partition names. If you specify a logical partition name for INITIAL\_ACCESS, you must also specify that same logical partition name in the ACCESS\_LIST list.

2. Devices attached to your system communicate with the system through control units and channel paths. Thus, before you dynamically define a new device to your system, you must first dynamically define one or more control units for that device (if they do not already exist) and one or more channel paths (if they do not already exist).

If you need to physically define any channel paths, control units, or devices, use the following commands in the following order:

- a. To define channel paths, use the DEFINE CHPID or PATH command.
- b. To define control units, use the DEFINE CU or CNTLUNIT command. (See “DEFINE CU / CNTLUNIT” on page 161.)
- c. To define devices, use the DEFINE DEVICE or IODEVICE command. (See “DEFINE DEVICE / IODEVICE” on page 167.)

If you do not define the channel paths, control units, and devices correctly and in the proper order, CP will issue appropriate error messages.

3. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the I/O configuration in the hardware system area

(HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.
  - b. Run the IOCP utility (see [“IOCP” on page 2203](#)) against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
  - c. Issue the SET IOCDS\_ACTIVE command (see [“SET IOCDS\\_ACTIVE” on page 1525](#)) to make the new IOCDS file the active file that the machine will use during the next power-on reset (POR).
4. For more information about defining channel paths in your I/O configuration, see [z/VM: I/O Configuration](#).
  5. To delete an existing channel path ID on the system, use the DELETE CHPID or PATH command. (See [“DELETE CHPID / PATH” on page 259](#).)
  6. To change the definition of an existing reconfigurable channel path ID on the system, use the MODIFY CHPID or PATH command. (See [“MODIFY CHPID / PATH” on page 679](#).) There is no MODIFY command to change the definitions of the other types of channel paths. If you need to change their definitions, you must:
    - a. Delete the existing channel path definition, using the DELETE CHPID or PATH command.
    - b. Redefine the channel path definition, using the DEFINE CHPID or PATH command.
  7. To display the status of a specific logical channel path and the devices with that logical channel path installed, or to display the type of a specific logical channel path, use the QUERY CHPID command. (See [“QUERY CHPID” on page 791](#).)
  8. To display the physical status of all of CP's defined channel paths, use the QUERY CHPIDS command. (See [“QUERY CHPIDS” on page 796](#).)
- Note:** If the channel path is in single channel service (SCS) mode, it will appear as "not configured" in the display.
9. To display the logical and physical status of the channel paths installed on one or more devices, use the QUERY PATHS command. (See [“QUERY PATHS” on page 1064](#).)
  10. On a machine with multiple logical channel-subsystem images, the DEFINE PATH and DEFINE CHPID commands define the channel path in the single specified channel-subsystem image. If the channel path is a type that can span multiple logical channel-subsystem images, then MODIFY PATH or MODIFY CHPID commands must be entered for each additional channel-subsystem image in which you want to define the channel path.
  11. When defining an OSD channel or an OSX channel, the user has the option of specifying whether or not to use the outbound priority queuing function supported by the QDIO interface. This option is similar to the CHPARM keyword on the IOCP CHPID statement for these types. If the option is exploited (PQ\_ON), then the maximum number of subchannels allowed on the defined OSD or OSX channel will be offset by the number of outbound priority queues. Since 3 subchannels are necessary per TCP/IP stack, 480 subchannels will yield a total of 160 stacks. 160 is the default maximum number of stacks with PQ\_ON.

Configurations that require many virtual machines to share the same OSD or OSX CHPID might need more than the default limit. When it is anticipated that more than 160 TCP/IP stacks will be required for a single OSD or OSX CHPID, then the PQ\_OFF option should be specified. By disabling the priority queues, the maximum number of subchannels increases to 1920. This allows 640 TCP/IP stacks to be supported per OSD or OSX CHPID. It should be noted that disabling priority queues may impact the performance of non-z/VM TCP/IP stacks and some z/VM virtual switch configurations that may be sharing subchannels on the OSD or OSX CHPID. This is because priority queues will no longer be available for their use. The z/VM TCP/IP stack does not make use of priority queues.

A CHPID defined for use by a z/VM Type IVL virtual switch must be configured with PQ\_ON to insure IVL network stability and availability. The IVL virtual switch exploits OSA-Express features priority queuing capabilities to insure IVL management transmissions are delivered to the IVL network ahead of any error recovery production traffic. In addition, a z/VM Type QDIO virtual switch can optionally

## DEFINE CHPID / PATH

exploit priority queuing by turning PRIQueuing ON. If you are planning on exploiting priority queuing on a z/VM virtual switch, then PQ\_ON must be specified.

**Note:** The outbound priority queuing function is always disabled (PQ\_OFF) for OSM channel paths.

### Responses

Response 1:

To define a switched ESCON channel path, enter the following:

```
define chpid 9 type escon switch 1
Channel path configuration change completed successfully
Ready;
```

Response 2:

To define a dedicated-reconfigurable FICON channel path, enter the following:

```
define chpid 1f type ficon dedicated_rec acc vm1 vm3 vm5 init vm1
Channel path configuration change completed successfully
Ready;
```

Response 3:

To define a shared channel path to an open systems adapter, enter the following:

```
define chpid 09 type osa shared access_list boston initial boston
Channel path configuration change completed successfully
Please remember to define a control unit and devices for an OSA
channel path
Ready;
```

Note that CP issues an extra message for OSA devices, because you should define the associated OSA control units and devices before leaving configuration mode.

Response 4:

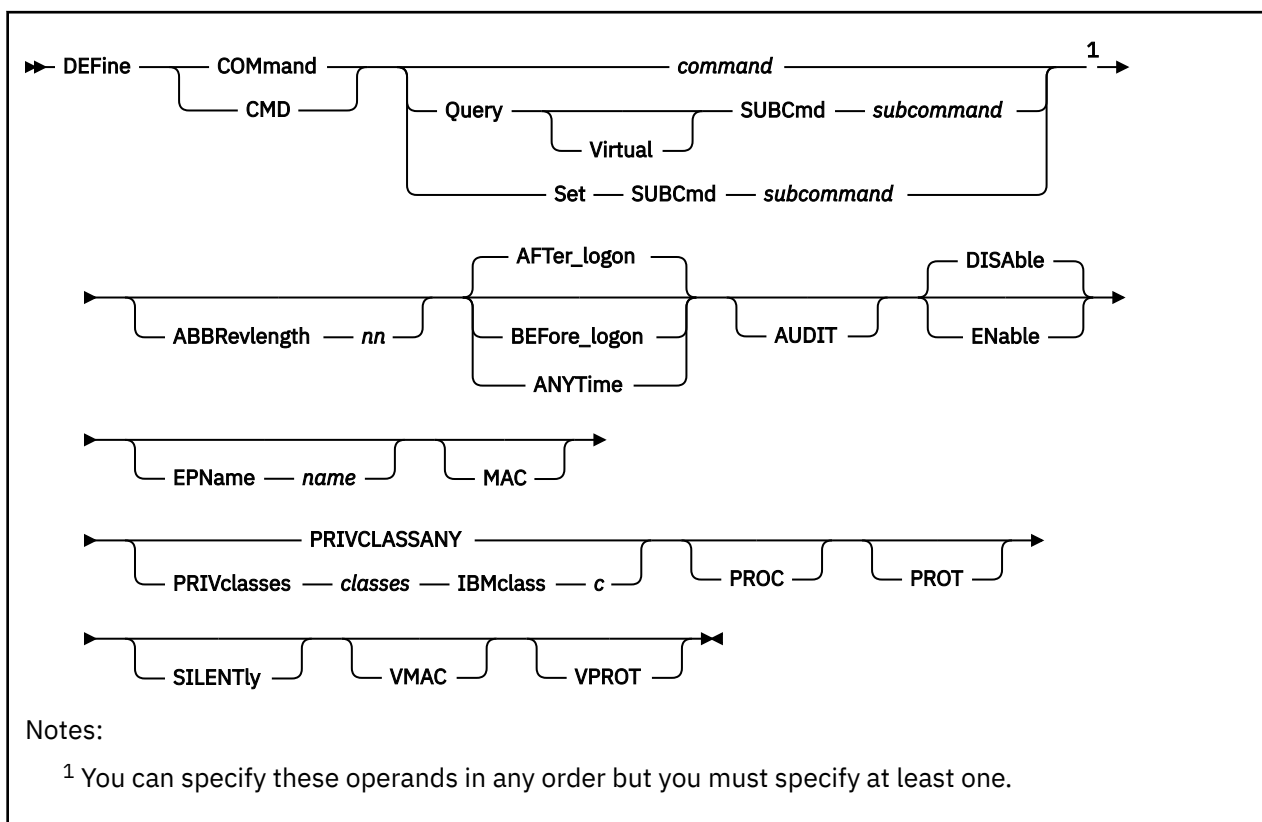
To define a shared FICON channel path in a channel-subsystem image, enter the following:

```
def chpid 43 pchid 81 css 2 type ficon shared acc vm1 vm2 vm3 init vm1
Channel path configuration change completed successfully
Ready;
```

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE COMMAND / CMD



### Authorization

Privilege Class: A

### Purpose

Use DEFINE COMMAND or DEFINE CMD to define a new CP command or a new version (by IBM class) of an existing CP command on the system.

You can also define a new CP command during initialization using the DEFINE COMMAND or DEFINE CMD configuration file statements. For more information, see [DEFINE COMMAND / CMD Statement in z/VM: CP Planning and Administration](#).

### Operands

#### **command**

is the name of the command that you are defining. The variable *command* is a 1- to 12-character string.

#### **Query SUBCmd *subcommand***

tells CP the name of the CP QUERY subcommand that you are defining. The variable *subcommand* is a 1- to 12-character string.

#### **Query Virtual SUBCmd *subcommand***

tells CP the name of the CP QUERY VIRTUAL subcommand that you are defining. The variable *subcommand* is a 1- to 12-character string.

### **Set SUBCmd *subcommand***

tells CP the name of the CP SET subcommand that you are defining. The variable *subcommand* is a 1- to 12-character string.

### **ABBRlength *nn***

is the length of the smallest acceptable abbreviation of the command or subcommand that you are defining. The variable *nn* is a decimal number between 1 and the length of the full command or subcommand name.

### **AFTer\_logon**

tells CP that the command version you are defining will only be issued by users after they log onto the system. Most CP commands fall into this category. The default is AFTER\_LOGON.

### **BEFore\_logon**

tells CP that the command you are defining will only be issued by users before they log onto the system. For example, the CP DIAL command can only be used before logon.

**Note:** If you specify BEFORE\_LOGON, you must specify PRIVCLASSANY because CP cannot check privilege classes before a user logs on. Thus, you cannot specify BEFORE\_LOGON with the PRIVCLASSES or IBMCLASS operands.

### **ANYTime**

tells CP that the command version you are defining can be issued by users both before and after they log onto the system. For example, many systems let users issue the CP MESSAGE or MSG commands before and after logon.

**Note:** If you specify ANYTIME, you must specify PRIVCLASSANY because CP cannot check privilege classes before a user logs on. Thus, you cannot specify ANYTIME with the PRIVCLASSES or IBMCLASS operands.

### **AUDIT**

tells the external security manager (ESM) to audit the command that you are defining. When you audit a command, the ESM logs each attempt by users to issue this command.

### **DISAble**

tells CP not to call the entry points and external symbols associated with this CP command until you enable it. (For more information about enabling CP commands, see Usage Note [“4”](#) on page 146.) If omitted, DISABLE is the default.

### **ENable**

tells CP to immediately start calling the entry points and external symbols associated with this CP command.

### **EPName *name***

tells CP the name of the entry point that contains the code to process the command. The variable *name* must be a 1- to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

**Note:** If you are defining a new command, you must specify EPNAME. If you are defining a new version of an existing command, specifying EPNAME is optional. This is because CP only allows one entry point per command, regardless of how many versions that command has. So, when you define a new version of an existing command, CP already knows the entry point name.

The QUERY and SET commands are the only exceptions, because they have subcommands. Note that CP only allows one entry point per subcommand, regardless of how many versions that subcommand has.

### **MAC**

tells CP to enable mandatory access control (MAC) for the command that you are defining. When MAC is enabled for your command, the external security manager (ESM) compares the security label of the user who issued your command to the security label of the resource or user that your command will affect. If you want the ESM to dynamically turn MAC on or off for this command, specify the VMAC operand.



**PRIVCLASSANY**

tells CP that users with any privilege class can issue the command that you are defining.

**PRIVclasses classes**

tells CP that only users with 1 or more of the specified privilege classes can issue the command that you are defining. The variable *classes* is 1 or more privilege classes in the range A through Z, 1 through 6, or an asterisk (\*). Privilege class \* indicates all privilege classes (A-Z and 1-6).

**Note:**

1. If you want more than one privilege class, specify your classes in one string of characters. Do not separate the classes with blank spaces. For example, specify "privclasses abc123", not "privclasses a b c 1 2 3".
2. If you specify PRIVCLASSES, you must also specify IBMCLASS. You can specify these 2 operands in any order.

**IBMclass c**

tells CP what type of command you are defining. The variable *c* can be any 1 of the following:

**A**

this is a system-control command to be used by the primary system operator.

**B**

this is a command for operational control of real devices.

**C**

this is a command to alter system storage.

**D**

this is a command for system-wide control of spool files.

**E**

this is a command to examine system storage.

**F**

this is a command for service control of real devices.

**G**

this is a general-use command used to control the functions of a virtual machine.

**Note:** If you specify IBMCLASS, you must also specify PRIVCLASSES. You can specify these 2 operands in any order.

**PROC**

tells CP that, after performing the initial privilege class checks, your command processor will be responsible for any further calls to the external security manager (ESM) for the command that you are defining.

**PROT**

tells the external security manager (ESM) to protect the command that you are defining. When protection is enabled for your command, the ESM checks an access list to ensure that the user who issued your command is authorized to do so. If you want the ESM to dynamically turn protection on or off for your command, specify the VPROT operand.

**SILENTly**

tells CP that the responses from the command you are defining can be suppressed by invoking it using the SILENTLY command. For more information, see [“SILENTLY” on page 1798](#).

**Note:** Response suppression is supported only for the ATTACH, DETACH, and GIVE commands.

**VMAC**

gives the external security manager (ESM) the power to dynamically turn mandatory access control (MAC) on or off. If you specify MAC and VMAC for this command, you are enabling MAC and allowing the external security manager (ESM) to dynamically disable MAC. If you specify VMAC and you do not specify MAC for this command, you are disabling MAC and allowing the ESM to dynamically enable MAC.



CP places no restriction on the use of special characters such as \*, %, comma, (, ), blank, /, :, \$, #, and @ in command names. However, some External Security Managers (ESMs) restrict their use. Please consult your ESM documentation for information on restricted characters.

11. If you try to define a minimum abbreviation that matches the abbreviation for an existing command, subcommand, or alias, CP rejects your DEFINE command. For example, if you created a "QUEUE" command with a minimum abbreviation of 2, CP would reject your QUEUE command because "QU" is an abbreviation for the QUERY command. In this case, you would need to specify a minimum abbreviation greater than or equal to 4 because the first 3 characters of QUERY and QUEUE are identical.
12. Once defined, commands, subcommands, aliases, and Diagnose codes cannot be deleted. They may be altered in various appropriate ways, but they remain in existence until a SHUTDOWN or RESTART IPL is done.
13. MODIFY COMMAND RESET will not allow a RESET to occur if the resulting command versions would have different EPNAMEs. This can occur if you change the EPNAME for a command then add a new version for that command. To avoid this restriction, you should add new versions of a command before changing the EPNAMEs associated with the command.
14. The DEFINE COMMAND or DEFINE DIAGNOSE commands should not be used on a system with an active External Security Manager (ESM); unpredictable results could occur. To safely define a CP command add the DEFINE COMMAND or DEFINE DIAGNOSE statement to the SYSTEM CONFIG file and re-IPL the system. This allows the command and diagnose additions to control structures to be made prior to activating the ESM.
15. For more information about user-defined commands, see [z/VM: CP Exit Customization](#).

## Responses

Response 1:

To define a new command, TELL, with:

- 3 versions:
  - Privilege class A, IBM class A, issued after logon
  - Privilege class B, IBM class B, issued after logon
  - Any privilege class, no IBM class, issued before and after logon,
- No minimum abbreviation, and
- The same entry point and syntax as the CP MESSAGE command,

enter the following:

```
define cmd tell epname hcpxmngms ibmclass a privclasses a
Ready;
define command tell ibmclass b privclasses b
Ready;
define command tell anytime privclassany
Ready;
```

**Important Note:** Normally, it is not a good practice to define a CP command that has the same name as a command, exec, or function belonging to another z/VM subsystem. (This newly-defined command will interfere with the CMS TELL command.) However, in this case, we thought we would show you an example of how to create a "synonym" for a common CMS command that could be used by novice z/VM users when they accidentally drop out of their CMS session into CP.

Response 2:

To define 2 new commands, CONTINUE and RESUME, which behave like the CP BEGIN command, enter the following:

```
define cmd continue abbr 4 epname hcpcmcbe ibmclass g priv g
Ready;
```

## DEFINE COMMAND / CMD

```
define command resume epname hcpcmcbe ibmclass g priv g  
Ready;
```

In this example, the CONTINUE and RESUME commands:

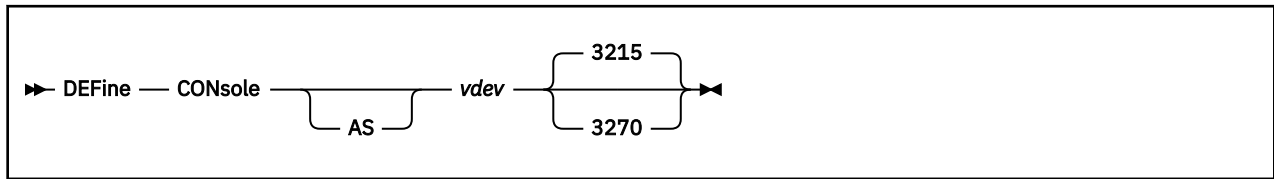
- Have the same entry point and syntax as the CP BEGIN command,
- Are IBM class G,
- Are privilege class G, and
- Can only be issued after logon.

Additionally, the CONTINUE command has a minimum abbreviation of 4 (CONT) and the RESUME command has no minimum abbreviation.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE CONSOLE



### Authorization

Privilege Class: G

### Purpose

Use DEFINE CONSOLE to add a system console to your virtual machine configuration.

### Operands

*vdev*

**AS** *vdev*

identifies the virtual device number of the console you are defining.

**3215**

**3270**

identifies the virtual console device type. Type 3215 is the default.

### Usage Notes

1. If a terminal is controlled by a VTAM service machine with a level lower than 3.1.1., the console cannot be defined as a 3270. If you attempt to do so, the console is defined as a 3215, and an error message is displayed.
2. Only one virtual console is permitted. You receive an error message if you already have a virtual console defined.
3. The default spool file class given to spool files created on the virtual console is class T.
4. Any time you enter the DEFINE CONSOLE command, you define the console mode for your display. This becomes the console mode for the duration of the session or until you reenter DEFINE CONSOLE or enter TERMINAL CONMODE. For more information on the console mode setting and its significance to the programs that you run in your virtual machine, see [“TERMINAL CONMODE” on page 1928](#).
5. For additional information, see [“Usage Notes” on page 122](#).

### Responses

Response 1:

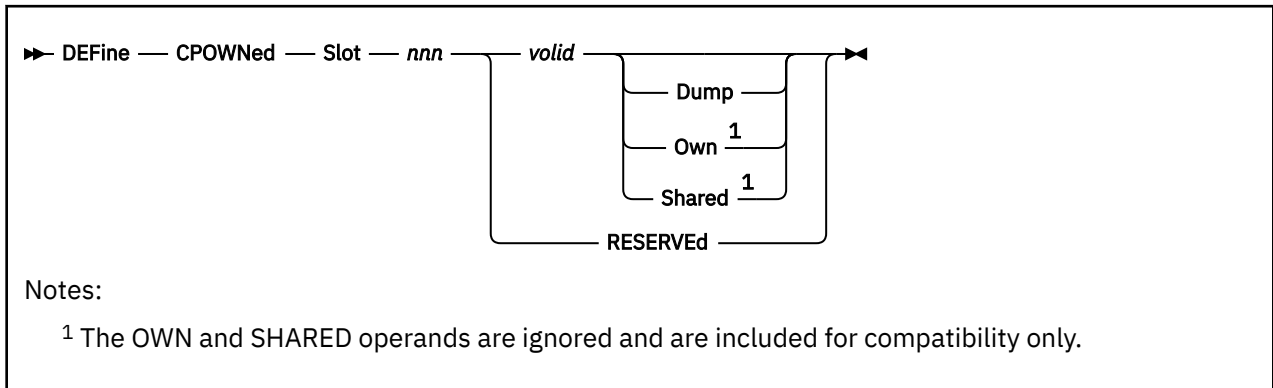
```
CONSOLE vdev DEFINED
```

confirms the virtual console definition.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE CPOWNEDED



### Authorization

Privilege Class: A

### Purpose

Use DEFINE CPOWNEDED to define new entries or to change existing entries in the list of CP-owned DASD volumes. You can also use the CP\_OWNED statement in the system configuration file to modify the CP-owned volume list. For more information on the CP\_OWNED statement, see the *z/VM: CP Planning and Administration* book.

### Operands

#### Slot *nnn*

specifies the number of the slot in the CP-owned volume list that you are defining or changing. The number *nnn* cannot exceed the number of slots available in the CP-owned volume list. You can use the QUERY CPOWNEDED command to find the number of slots that were defined at IPL time. For more information on the QUERY CPOWNEDED command, see [“QUERY CPOWNEDED”](#) on page 834.

#### *valid*

specifies the volume to add to the CP-owned volume list. You cannot specify the same volume identifier in more than one slot in the CP-owned volume list.

#### Dump

tells CP to reserve the spool space on the specified volume exclusively for dumps.

#### Own

is ignored and is included for compatibility only. In an SSI cluster, volume ownership is determined by the ownership information recorded on the volume.

#### Shared

is ignored and is included for compatibility only. In an SSI cluster, volume ownership is determined by the ownership information recorded on the volume.

#### RESERVED

tells CP to disassociate the slot from a specific volume and make it available for allocation.

### Usage Notes

1. Unless the system configuration has been updated, all definitions from the DEFINE CPOWNEDED commands will be lost at the next system initialization. Always consider updating the system configuration file before issuing the DEFINE CPOWNEDED commands.
2. Use the QUERY CPOWNEDED command to display the CP-owned volume list and the number of available slots. For more information, see [“QUERY CPOWNEDED”](#) on page 834.

3. You cannot replace an existing entry in the CP-owned volume list if the volume being replaced has not been detached from the system. For more information, see the DETACH (Real Device) command.
4. A volume can be defined as a CP-owned volume only if it is not currently attached to SYSTEM.
5. A slot may not be changed to reserved status unless the volume associated with the slot is not attached to the system.
6. In an SSI cluster, when a volume with SPOOL extents is defined as CP-owned on a member of the cluster and the ATTACH command is issued, the attach process will automatically attach the volume to the other active members where the device is online and not already attached. Also, if the volume is not in the CP-owned list on another active member where the device is being attached, the attach process will automatically add the volume to the CP-owned list on that member.
7. See *z/VM: CP Planning and Administration* for information on allocating space for CP hardabend dumps including recommendations from IBM.

## Responses

Response 1:

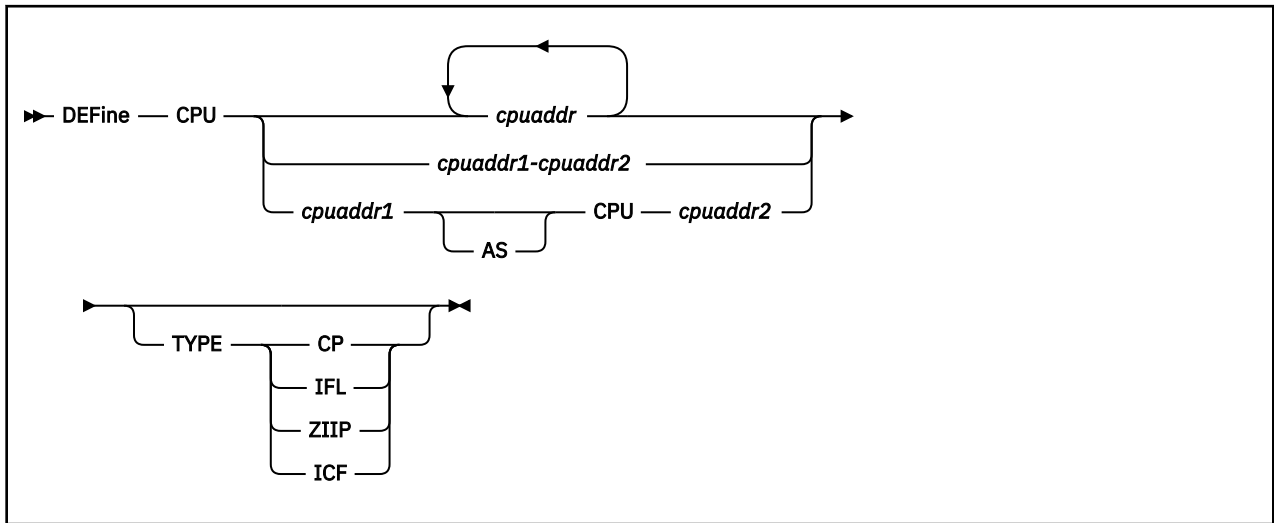
To add the shared volume ESAPG1 to the CP-owned DASD volume list in the empty slot 10, enter:

```
define cowned slot 10 esapg1 shared
Ready;
```

## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE CPU



### Authorization

Privilege Class: G

### Purpose

Use DEFINE CPU to do the following:

- Add one or more processors to your virtual machine
- Change the address of a defined virtual processor.
- Change the type of a defined virtual processor.

### Operands

#### **cpuaddr**

defines a virtual processor for your virtual machine configuration at the address you specify for *cpuaddr*. You can specify multiple addresses in a list or in range form. If you specify a list of addresses, separate each address from the others by at least one blank. If you specify a range of addresses, specify the first and last addresses in the range and put a hyphen between them. The processor addresses can be any value from 0 to 3F; the initial address of your base processor is specified in your system directory entry, or it defaults to 0.

#### **cpuaddr cpuaddr**

identifies a list of addresses or ranges delimited by blanks.

#### **cpuaddr1-cpuaddr2**

specifies a nonwrapping pair of addresses separated by a hyphen (-) and not separated by any blanks.

#### **cpuaddr1 AS CPU cpuaddr2**

changes the address of the processor currently defined at *cpuaddr1* to the address *cpuaddr2*.

#### **TYPE**

specifies the type of virtual CPU being defined. If no type is specified when defining a new CPU, it defaults to the type of the primary virtual CPU.

If no type is specified when redefining an existing virtual CPU (for example, changing the CPU address), the type of the virtual CPU does not change. Each command to define or redefine virtual CPU(s) must result in a valid virtual CPU configuration. Valid types are:

#### **CP**

indicates the central processor.



**IFL**

indicates the IBM Integrated Facility for Linux.

**ZIIP**

indicates the IBM z Integrated Information Processor.

**ICF**

indicates the Internal Coupling Facility.

Valid virtual configurations are shown in the tables below:

*Table 13. ESA/390 or General Logical Partition*

Primary Real CPU Type	VCONFIG Mode	Primary Virtual CPU Type	Valid Virtual CPU Types
CP	ESA390, GENERAL ***	CP	CP, zIIP **
CP	CF *	CP	CP
CP	LINUX	IFL	IFL
CP	LINUX	CP	CP
CP	VM	CP	CP, zIIP, IFL, ICF **

*Table 14. LINUX Only Logical Partition*

Primary Real CPU Type	VCONFIG Mode	Primary Virtual CPU Type	Valid Virtual CPU Types
CP	LINUX	CP	CP
CP	LINUX	IFL	IFL
IFL	LINUX	IFL	IFL

*Table 15. z/VM Logical Partition*

Primary Real CPU Type	VCONFIG Mode	Primary Virtual CPU Type	Valid Virtual CPU Types
CP	ESA390, GENERAL ***	CP	CP, zIIP **
CP	CF *	ICF	ICF
CP	LINUX	IFL	IFL
CP	LINUX	CP	CP
CP	VM	CP	CP, zIIP, IFL, ICF **

where: \* indicates OPTION CFVM was specified in the user directory entry

\*\* indicates that zIIPs are supported as secondary virtual CPU types as long as they are supported on the underlying machine model

\*\*\* VCONFIG modes ESA390 and GENERAL are functionally equivalent, which means that the same virtual CPU types can be defined in an ESA390 mode virtual machine and in a GENERAL mode virtual machine

**Usage Notes**

1. Your system directory entry specifies the maximum number of processors you can define for your virtual machine.

## DEFINE CPU

2. If your system directory entry contains multiple CPU statements and/or DEFINE CPU commands (on directory COMMAND statements) that specify a virtual multiprocessing configuration, that configuration is created for you at logon time. If your directory entry does not specify additional processors, your virtual machine logs on as a uniprocessor. Once logged on, you can use DEFINE CPU to modify your virtual configuration.
3. Any CPUs defined with the CPU statement in your directory entry will default to the same type as your primary real processor.
4. The following actions will result in a System Clear:
  - Changing the type of an existing CPU.
  - Changing the address of a CPU in a multi-CPU-type virtual configuration.
5. Each DEFINE CPU command must result in a valid virtual CPU configuration. Because of this, the order of multiple DEFINE CPU commands is important.
6. To make changes to a virtual configuration, including changing virtual CPU types and the virtual configuration mode, the best approach is to:
  - a. Issue the DEFINE CPU command to change all CPU types to CP.
  - b. Issue the SET VCONFIG MODE command to change the virtual configuration mode.
  - c. Issue the DEFINE CPU command to change the CPU types to the desired types.
7. The virtual configuration mode of a Coupling Facility virtual machine cannot be changed.
8. All future CP responses are prefaced with the number of the CPU that issues the responses. This continues as long as you have the virtual multiprocessor configuration.
9. A user that is assigned to a resource pool cannot change the CPU type of its base CPU because the base CPU defines the user's primary CPU type, and the primary CPU type must match the CPU type being limited by the resource pool. These CPU types matched when the user was put into the resource pool and must continue to do so. As long as the user's logical configuration remains valid, a user assigned to a resource pool can define additional CPUs or change the type of the non-base CPUs. The QUERY VIRTUAL CPUS command can be used to determine the user's base CPU number and type.
10. For additional information, see ["Usage Notes" on page 122](#).

## Responses

Response 1:

```
xx: CPU nn DEFINED
```

confirms that the processor has been defined.

Response 2:

```
xx: CPU nn REDEFINED AS TYPE yyyy
```

confirms that the processor has been redefined to a new type.

**xx**

identifies the number of the CPU responding to you.

**nn**

identifies the number of the CPU currently defined or redefined.

**yyyy**

identifies the type of the redefined CPU.

Response 3:

```
xx: STORAGE CLEARED - SYSTEM RESET
```

indicates that a system-reset-clear has been performed.

## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

Messages:

- HCP5780E Adjunct does not support the DEFINE CPU command
- HCP6849E DEFINE CPU cannot change the CPU type of the base CPU for a userid in a resource pool

## DEFINE CPUPOOL

---

### **Purpose**

DEFINE CPUPOOL has been superseded by DEFINE RESPOOL. See [“DEFINE RESPOOL” on page 205](#).

## DEFINE CRYPTO

►► DEFine — CRYPTo — APVIRTual ◄◄

### Authorization

Privilege Class: G

### Purpose

Use DEFINE CRYPTO APVIRTUAL command to create a virtualized crypto resource through which the virtual machine can use the system's pool of shared crypto resources.

### Usage Notes

1. The DEFINE CRYPTO APVIRTUAL command requires a virtual machine to be enabled to access the shared crypto resources. A CRYPTO APVIRTUAL directory statement must be included in the directory entry of the virtual machine requesting shared use of crypto resources.
2. The DEFINE CRYPTO APVIRTUAL command allows a general user to regain access to the system's shared crypto resources after the virtual resource has been detached.
3. Use the DETACH CRYPTO APVIRTUAL command to remove a virtual machine's access to the system's shared crypto resources.
4. If DEFINE CRYPTO APVIRTUAL is issued when there are no crypto resources in the shared pool, the virtual machine will still be given access to the system's shared crypto resources if they become available.
5. For information on planning and managing crypto resources on a z/VM system, see [Crypto Planning and Management in z/VM: CP Planning and Administration](#).

### Responses

Response 1:

The following is the response if there are resources in the shared crypto pool.

```
Crypto APVIRTUAL defined
```

Response 2:

The following is the response if there are no resources in the shared crypto pool.

```
Crypto APVIRTUAL defined, however, there are
currently no shared AP crypto resources available.
```

### Messages

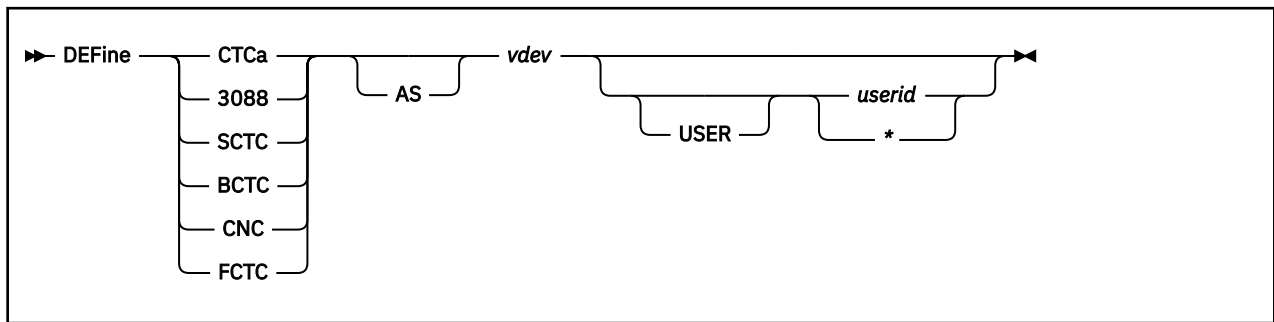
For the list of general messages for the DEFINE command, enter HELP CP DEFINE.

- HCP003E Invalid option - *option*
- HCP003E Invalid option - *option*| command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP052E Error in CP directory
- HCP1014E A required option is missing
- HCP1724E Crypto APVIRTUAL is already defined.

## DEFINE CRYPTO

- HCP1745E Crypto APVIRTUAL cannot be defined because you do not have authorization to access shared crypto resource on this system.
- HCP1746E Crypto APVIRTUAL cannot be defined because you already have access to dedicated crypto resources.

## DEFINE CTCA



### Authorization

Privilege Class: G

### Purpose

Use DEFINE CHANNEL-TO-CHANNEL ADAPTER (CTCA) to define a CTCA for your virtual machine configuration.

### Operands

#### CTCa

indicates you are defining a channel-to-channel adapter for your virtual machine.

#### 3088

indicates you are defining a channel-to-channel adapter for your virtual machine to emulate a 3088 device.

#### SCTC

indicates you are defining a channel-to-channel adapter for your virtual machine to emulate an ESCON CTCA in Standard mode.

#### BCTC

indicates you are defining a channel-to-channel adapter for your virtual machine to emulate an ESCON CTCA in Basic mode.

#### CNC

indicates you are defining a channel-to-channel adapter for your virtual machine to emulate an ESCON CTCA with no controller (the CNC device must be coupled to an SCTC or BCTC adapter to be fully functional).

#### FCTC

indicates you are defining a channel-to-channel adapter for your virtual machine to emulate a FICON CTCA.

#### *vdev*

#### **AS** *vdev*

identifies the virtual device number of the CTCA you are defining.

#### *userid*

#### **USER** *userid*

identifies the user that you will permit to connect to this CTCA using the COUPLE command. If you do not specify *userid*, you are not restricting any user ID from connecting.

#### \*

can be used to specify your own user ID.

### Usage Notes

1. For general usage notes for the DEFINE COMMAND, see [“Usage Notes” on page 122](#).

### Responses

Response 1:

```
CTCA vdev DEFINED
```

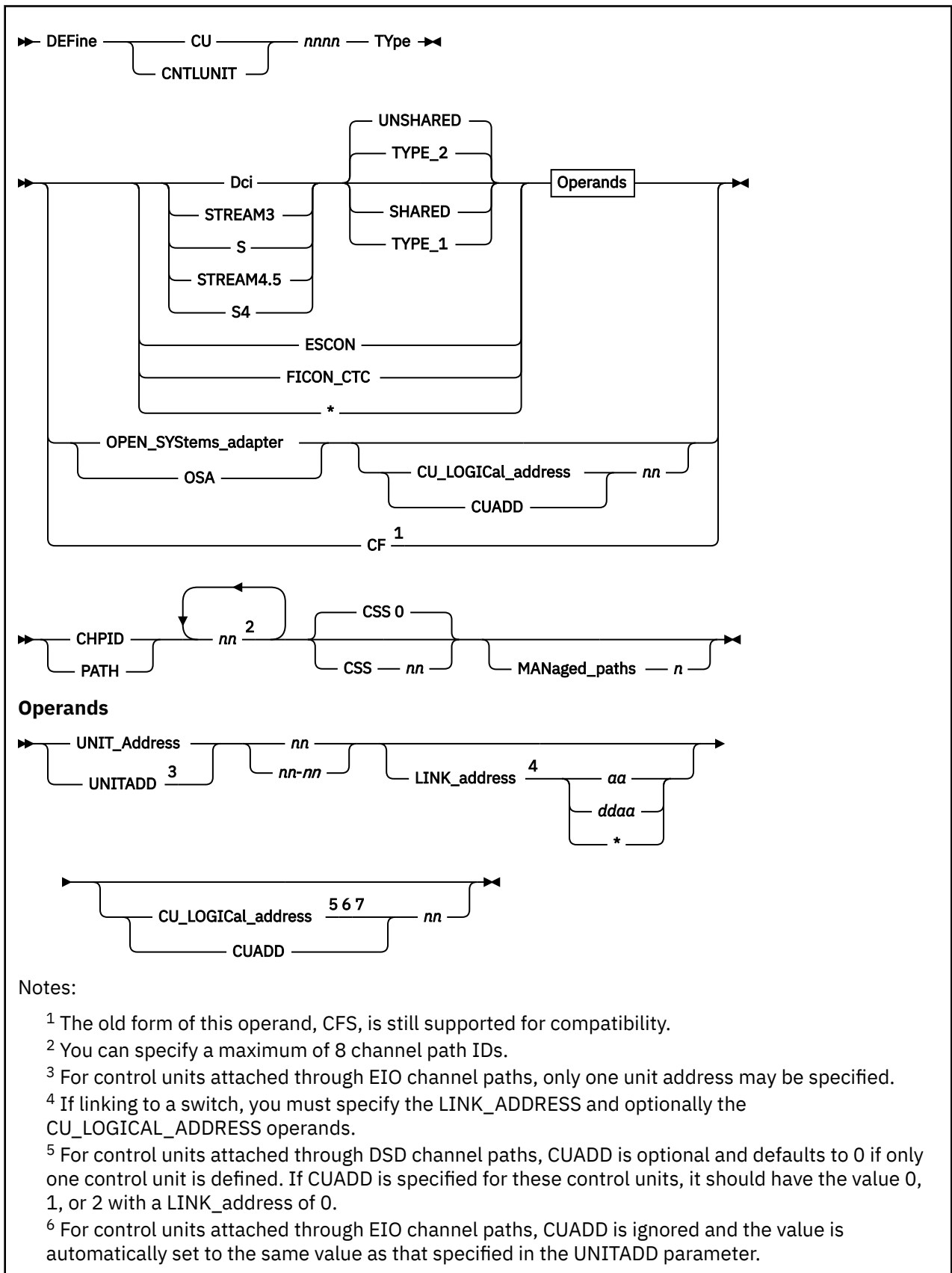
confirms the definition of the virtual channel-to-channel adapter.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).



## DEFINE CU / CNTLUNIT



<sup>7</sup> For FICON CTCs on a shared channel path, the CUADD parameter specifies the MIF-ID number of the destination partition; that is, the remote end of the CTC connection for the control unit being defined. Use the QUERY LPAR command to find the MIF-ID for the LPAR. A control unit must be defined for each destination partition on the shared channel path. For FICON CTCs on non-shared channel paths, the destination partition is implicit in the channel-path definition and should not be specified here.

## Authorization

Privilege Class: B

## Purpose

Use DEFINE CU or CNTLUNIT to define a new control unit to the I/O configuration.

**Note:** DEFINE CU and DEFINE CNTLUNIT are dynamic I/O commands. Dynamic I/O configuration is only supported on IBM Z servers. If your system is not running on one of these servers, your command will not complete successfully.

## Operands

### *nnnn*

is the number of the control unit that you are defining. The variable *nnnn* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFE'.

### **T**Ype

is the control unit type. Valid types are:

### **D**ci

tells CP that this control unit uses the directed-coupled interlock (DCI) protocol. DCI protocol is a method of data transfer up to 1.5 megabytes per second. DCI protocol requires the sender to raise a signal on the interface along with the byte of data being transferred. The signal and data must be maintained until the receiver acknowledges receipt of the data.

### **S**TREAM3

#### **S**

tells CP that this control unit uses a method of data transfer up to 3 megabytes per second. This protocol does not interlock data streaming between sender and receiver. Once the data transfer begins, the sender does not wait for acknowledgment from the receiver before sending the next byte of data. The control unit determines the actual data transfer rate.

### **S**TREAM4.5

#### **S**4

tells CP that this control unit uses a method of data transfer up to 4.5 megabytes per second. This protocol does not interlock data streaming between sender and receiver. Once the data transfer begins, the sender does not wait for acknowledgment from the receiver before sending the next byte of data. The control unit determines the actual data transfer rate.

### **U**NSHARED

#### **T**YPE\_2

tells CP that this control unit supports concurrent multiple I/O requests. This means that this control unit is synonymous with a Type 2 control unit.

### **S**HARED

#### **T**YPE\_1

tells CP that this control unit only supports one I/O request at a time. This means that this control unit is synonymous with a Type 1 control unit.

**ESCON**

tells CP that this control unit uses the Enterprise Systems Connection (ESCON) channel to control unit I/O interface to transmit data through optical cables. Also specify this operand for control units attached through direct-system-device (DSD), emulated-I/O (EIO), FICON converter (FCV) or internal-system-device (ISD) channel paths.

**FICON\_CTC**

tells CP that the control unit is a channel-to-channel (CTC) adapter on a FICON channel path. At least one of the two sides of a FICON CTC connection must have a FICON CTC control unit defined. If a FICON CTC control unit is defined on both sides (which is recommended), the two connected channels determine which side acts as the control unit at any given time. For a FICON CTC control-unit definition to be effective, the FICON CTC facility must be installed in the host with the control-unit definition

**\***

indicates that the correct TYPe will be chosen automatically. Automatic selection applies to control units attached through DSD, EIO, CNC, CVC, CBY, CTC, FC, FCP, FCV, IQD, ISD, OSC, or OSN channel paths.

**OPEN\_SYSTEMS\_adapter****OSA**

tells CP that this control unit connects to an Open Systems Adapter (OSA), OSA Direct Express (OSD), OSA-Express (OSE), OSA Direct Express IEDN (OSX), or OSA Direct Express INMN (OSM) channel path.

**Notes:**

1. Before you have a working OSA configuration, you should define the channel paths, the device, and the control unit. See the *Planning for the System/390 Open Systems Adapter Feature GC23-3890*.
2. Only 1 CU per OSA CHPID per logical partition is allowed.

**CF**

tells CP that this control unit connects to coupling facility channel paths. Coupling facility channel paths include coupling facility sender (CFS), coupling facility receiver (CFR), coupling facility peer (CFP), cluster bus sender (CBS), cluster bus receiver (CBR), cluster bus peer (CBP), or internal coupling peer (ICP) channel paths.

**UNIT\_Address nn****UNIT\_Address nn-nn****UNITADD nn****UNITADD nn-nn**

is the last 2 digits of a device address or number (or a range of addresses or numbers). Each *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**LINK\_address aa****LINK\_address ddaa****LINK\_address \***

Each link-address value you specify identifies a port on a destination switch (ESCON or FICON director) to which the control unit is connected.

The LINK\_address operand must be specified when one or more of the channel paths to the control unit are switched paths; that is, when any of the paths traverse an ESCON or FICON director and for an ESCON director the path through the director is not dedicated. When all of the channel paths are unswitched (including dedicated paths through an ESCON director), the LINK\_address operand is optional.

For an unswitched FICON channel path, the end nodes determine the link addresses used at each end. Thus you cannot specify a numeric link address to be used. If you specify a LINK\_address operand value for an unswitched FICON channel path (for example, because other paths to the control unit are switched and a "placeholder" value is required), you must specify an asterisk (\*) as the link-address value.

When any link addresses are required to be specified (for example, when one or more of the channel paths is a switched path), you must provide a link address for each channel-path identifier (PATH or CHPID) that you specified, in the same order. When no link addresses are required to be specified

(i.e., when all of the channel paths are unswitched paths), the link address for each ESCON channel path is determined by the hardware processor, and the link address for each FICON channel path is negotiated between the hardware endpoints.

The link address is a one or two digit hexadecimal value for control units connected to an ESCON channel. The link address is a one, two, three, or four digit hexadecimal value for control units connected to a FICON channel. Leading zeroes can be omitted, except that an actual *dd* value of zero (see below) must be specified with at least one digit.

**aa**

is the one-byte port-area address in an ESCON or FICON director to which the control unit is connected. This address must be a one or two digit hexadecimal number between X'01' and X'FE' for ESCON directors, and between X'00' and X'FF' for FICON directors.

Link addresses for channel paths having dedicated (unswitched) connections through an ESCON director can be arbitrary within the above-specified range, but it is recommended that the actual port-area address be used.

**ddaa**

is the first two bytes of a standard three-byte fibre-channel port address. The third byte of a standard port address is used for fibre-channel arbitrated loops, which are not supported for FICON channels and therefore you do not specify. This two-byte form of the link address is required when the channel path is a FICON channel path that traverses more than one FICON director to reach any of the channel-path's destination control units.

This means that if the channel path traverses more than one FICON director to reach this control unit, a two-byte link address is required. In addition, if the channel path traverses only one FICON director to reach this control unit, but the same channel path traverses more than one FICON director to reach some other control unit, a two-byte link address is required for that channel path in all control-unit definitions in which the channel path appears. That is, once a two-byte link address is required for any control unit on a channel path, a two-byte link address is required for all control units on that channel path.

**dd**

is the switch address (or domain identifier) of the destination FICON director (for example, the switch to which the control unit is directly connected), and *aa* is the port-area address within that FICON director. The link address must be a three to four digit hexadecimal number between X'0000' and X'FFFF'.

One-byte and two-byte link addresses can be mixed within the LINK\_address operand, according to the requirements for each channel path. You can choose to use a two byte link address when only a one byte address is required as long as the FICON director is configured with the high-integrity attributes (see the *IOCP User's Guide* for more information).

**CU\_LOGICAL\_address nn**

**CUADD**

is the logical address of the control unit that you are defining. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FE'. If omitted, the default is X'00'.

For control units attached through DSD channel paths, CUADD is optional and defaults to 0 if only one control unit is defined. If CUADD is specified for these control units, it should have the value 0, 1, or 2 with a LINK\_address of 0.

For control units attached through EIO channel paths, CUADD is ignored and the value is automatically set to the same value as that specified in the UNITADD parameter.

For control units attached through CNC, CTC, FCV, or IQD channel paths, the variable *nn* must be a 1 to 2-digit hexadecimal number between X'00' and X'0F'.

**CHPID nn**

**PATH nn**

is the value that uniquely identifies the path from this control unit to the system. You can specify up to 8 channel path IDs (CHPIDs). Each *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'. All channel paths must be in the same channel-subsystem image. .

**Note:** You cannot specify managed channel paths.

### CSS *nn*

is the channel-subsystem ID of the specified channel paths. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'.

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require the logical channel-subsystem facility to be installed.

### MANaged\_paths *n*

is the maximum number of managed channel paths that can be attached to the specified control unit. The variable *n* must be a 1-digit decimal number from 1 to 7.

**Note:** The actual managed paths cannot be specified in the control-unit definition; only the maximum number of managed channel paths can be specified. The managed paths will be associated with specific control units automatically by z/OS's dynamic-channel-path-management function. When a maximum number of managed paths is specified, the channel paths named in the CHPID/PATH operand must all be shared and of the type CNC, FC, or FCV. Also, the number of paths specified with the CHPID/PATH operand when added to the maximum number of managed paths specified cannot exceed 8.

## Usage Notes

1. Devices attached to your system communicate with the system through control units and channel paths. Thus, before you dynamically define a new device to your system, you must first dynamically define one or more control units for that device (if they do not already exist) and one or more channel paths (if they do not already exist).

If you need to physically define any channel paths, control units, or devices, use the following commands in the following order:

- a. To define channel paths, use the DEFINE CHPID or PATH command. (See [“DEFINE CHPID / PATH”](#) on page 132.)
- b. To define control units, use the DEFINE CU or CNTLUNIT command.
- c. To define devices, use the DEFINE DEVICE or IODEVICE command. (See [“DEFINE DEVICE / IODEVICE”](#) on page 167.)

If you do not define the channel paths, control units, and devices correctly and in the proper order, CP will issue appropriate error messages.

2. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.
- b. Run the IOCP utility against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
- c. Issue the SET IOCDS\_ACTIVE command to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR).

For more information, see [“IOCP”](#) on page 2203 and [“SET IOCDS\\_ACTIVE”](#) on page 1525.

3. For more information about defining control units in your I/O configuration, see [z/VM: I/O Configuration](#).
4. To change the definition of an existing control unit on the system, use the MODIFY CU or CNTLUNIT command. (See [“MODIFY CU / CNTLUNIT”](#) on page 688.)
5. To delete an existing control unit on the system, use the DELETE CU or CNTLUNIT command. (See [“DELETE CU / CNTLUNIT”](#) on page 263.)

6. On a machine with multiple logical channel-subsystem images, the DEFINE CU and DEFINE CNTLUNIT commands define the control unit on channel paths within the single specified channel-subsystem image. If the control unit is connected to channel paths in more than one channel-subsystem image, then MODIFY CU or MODIFY CNTLUNIT commands must be entered to define the control unit's connections to each group of channel paths in the additional channel-subsystem images.

### Responses

Response 1:

To define an Open Systems Adapter (OSA) control unit, enter the following:

```
def cu 1210 type OSA chpid 42
Control unit configuration change completed successfully
Ready;
```

Response 2:

To define an ESCON control unit, enter the following:

```
def cu 900 type escon unit_addr 0-1f link_addr f cu_logic 9 chpid 9
Control unit configuration change completed successfully
Ready;
```

Response 3:

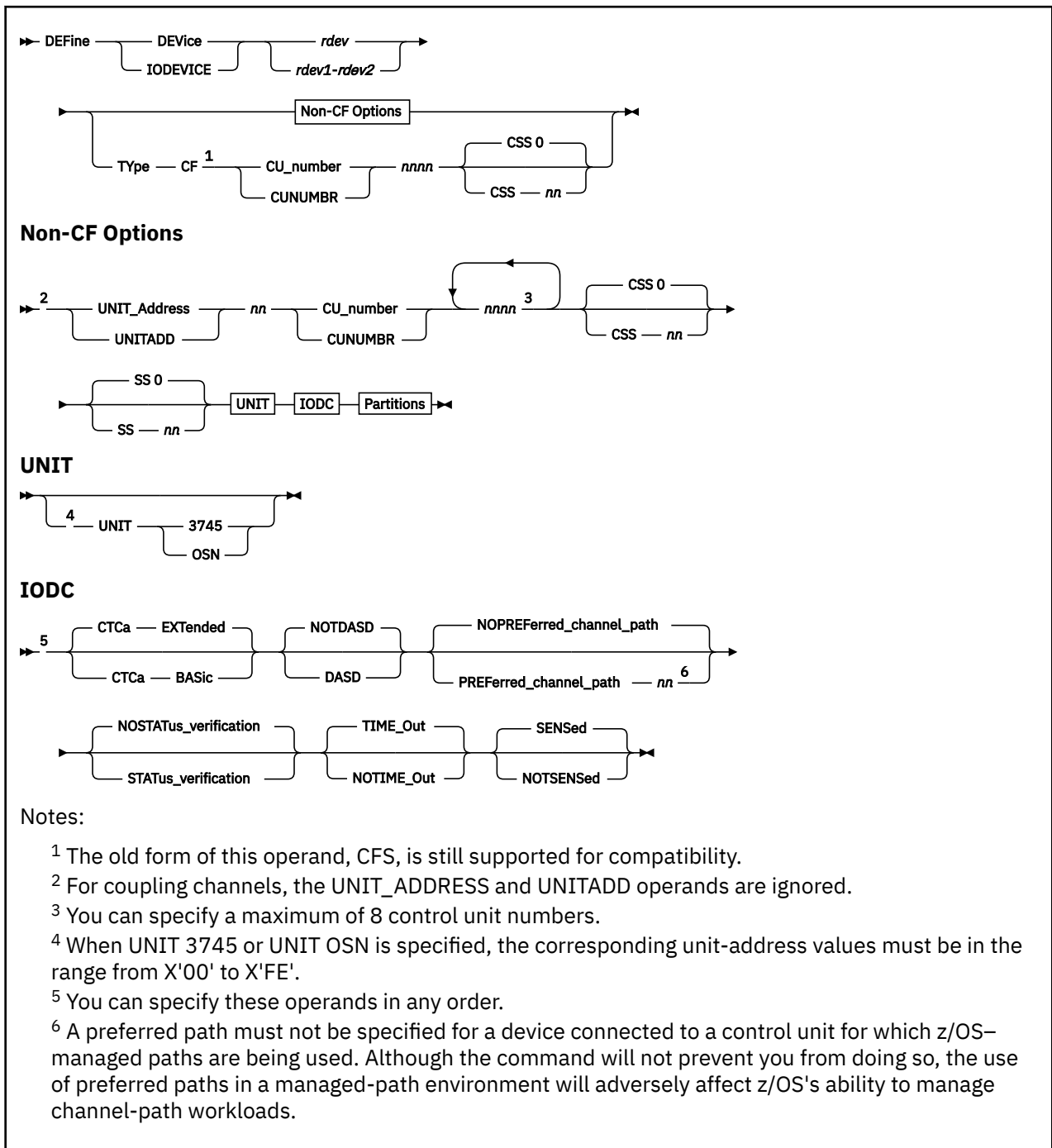
To define a FICON control unit to a channel-subsystem image, enter the following:

```
def cu 4343 type * unitadd 81 chpid 7 9 42 css 1
Control unit configuration change completed successfully
Ready;
```

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE DEVICE / IODEVICE



**Partitions**

►► PARTition — *pname* <sup>1</sup>►►

Notes:

<sup>1</sup> The maximum number of logical partition names you can specify depends on the capabilities of your machine.

**Authorization**

Privilege Class: B

**Purpose**

Use DEFINE DEVICE or IODEVICE to define one or more new real devices to the I/O configuration.

**Operands**

*rdev*

*rdev1-rdev2*

is the real device number (or range of real device numbers) of the device that you are defining to CP. Each *rdev* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFF'.

**UNIT\_Address *nn***

**UNITADD *nn***

is the last 2 digits of a device address or number. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**Note:** If you are specifying a range of devices (*rdev1-rdev2*), the unit address that you specify is the starting address associated with the first device in the range.

**TType CF**

tells CP that the specified devices are attached to a control unit that connects to coupling facility channel paths. Coupling facility channel paths include coupling facility sender (CFS), coupling facility receiver (CFR), coupling facility peer (CFP), cluster bus sender (CBS), cluster bus receiver (CBR), cluster bus peer (CBP), or internal coupling peer (ICP) channel paths.

**CU\_number *nnnn***

**CUNUMBR *nnnn***

is the number of the control unit associated with the device that you are defining. You can specify up to 8 control unit numbers. Each *nnnn* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFE'.

**CSS *nn***

is a channel-subsystem ID previously defined for one or more channel paths to the specified control unit. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'.

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require that the logical channel-subsystem facility be installed.

**SS *nn***

is the subchannel set with which this device is associated. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'03'.

**Note:** This parameter requires that the multiple-subchannel-set facility be installed.

**UNIT 3745**

tells CP that this device is a simulated 3745 device.



**UNIT OSN**

tells CP that this device is an OSA NCP device.

**Note:** When UNIT 3745 or UNIT OSN is specified, the corresponding unit-address values must be in the range from X'00' to X'FE'.

**CTCa EXTended**

tells CP that this device is an extended mode channel-to-channel adapter (CTCA).

**CTCa BASic**

tells CP that this device is a basic mode channel-to-channel adapter (CTCA).

**NOTDASD**

tells CP that this device is not a direct access storage device (DASD).

**DASD**

tells CP that this device is a direct access storage device (DASD).

**NOPREferred\_channel\_path**

tells CP that there is no preferred channel path.

**PREFerred\_channel\_path *nn***

tells CP to use this channel path to initiate I/O requests before trying any other channel paths from the device that you are defining. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**Note:** A preferred path must not be specified for a device connected to a control unit for which z/OS-managed paths are being used. Although the command will not prevent you from doing so, the use of preferred paths in a managed-path environment will adversely affect z/OS's ability to manage channel-path workloads.

**NOSTATus\_verification**

tells CP that the status verification facility is not active for this device, and so the channel subsystem will not check the device's status.

**STATus\_verification**

tells CP that the status verification facility is active for this device, causing the channel subsystem to check the device's status every time this device announces its status.

**NOTIME\_Out**

tells CP that the I/O interface time out function is not active. This parameter should not be specified if the device is being defined on a CHPID with TYPE=BYTE.

**TIME\_Out**

tells CP that the I/O interface time out function is active for the following I/O interfaces between the channel and I/O device:

- 4 second time out for initial selection
- 30 second time out for data transfer.

**SENSEd**

tells CP to use the information returned from a sense ID request to determine the device class or type at device initialization.

**NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the device class or type at device initialization.

**PARTition *pname***

specifies the name(s) of the logical partition(s) which will have access to the device(s) being defined in the specified or implied channel-subsystem image. The maximum number of logical partition names you can specify depends on the capabilities of your processor. Each *pname* must be a 1- to 8-character alphanumeric string.

**Note:** When defining CTC devices, only specify a PARTition list that includes partitions in which they will be used. If "extra" partitions are specified, then the CTC subchannels defined for those extra partitions unnecessarily use Hardware System Area (HSA) storage.

## Usage Notes

1. Devices attached to your system communicate with the system through control units and channel paths. Thus, before you dynamically define a new device to your system, you must first dynamically define one or more control units for that device (if they do not already exist) and one or more channel paths (if they do not already exist).

If you need to physically define any channel paths, control units, or devices, use the following commands in the following order:

- a. To define channel paths, use the DEFINE CHPID or PATH command. (See [“DEFINE CHPID / PATH” on page 132.](#))
- b. To define control units, use the DEFINE CU or CNTLUNIT command. (See [“DEFINE CU / CNTLUNIT” on page 161.](#))
- c. To define devices, use the DEFINE DEVICE or IODEVICE command.

If you do not define the channel paths, control units, and devices correctly and in the proper order, CP will issue appropriate error messages.

2. Before you add a new device to the system, you should decide whether you want CP to immediately initialize that device. If CP initializes a new device when you add it to the system, you can begin using that device immediately. If CP does not initialize the new device, you must vary the device online using the VARY (real device) command (see [“VARY \(Real Device\)” on page 2063](#)) before anyone can use that device.
3. To display whether CP initializes new real devices as they are added to a running system, use the QUERY NEW\_DEVICES command. (See [“QUERY NEW\\_DEVICES” on page 1033.](#))
4. To define or change the way CP initializes new real devices as they are added to a running system:
  - During initialization, use the ENABLE or DISABLE NEW\_DEVICES\_INITIALIZED\_WHEN\_ADDED operands of the FEATURES statement in your system configuration file. For more information, see [FEATURES Statement](#) in *z/VM: CP Planning and Administration*.
  - After initialization, use the SET NEW\_DEVICES command. (See [“SET NEW\\_DEVICES” on page 1581.](#))
5. To change whether CP uses the sense ID information at device initialization, use the SENSED or NOTSENSED operands of the SET DEVICES command. (See [“SET DEVICES” on page 1490.](#))
6. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.
- b. Run the IOCP utility against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
- c. Issue the SET IOCDS\_ACTIVE command to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR).

For more information, see [“IOCP” on page 2203](#) and [“SET IOCDS\\_ACTIVE” on page 1525.](#)

## Responses

Response 1:

To define 16 real DASDs in two logical partitions with a preferred channel path, enter the following:

```
def dev 300-30f unit_addr 0 cu 300 dasd preferred 3 par vm1 vm5
Device configuration change completed successfully
Ready;
```

Response 2:

To define 16 real DASDs in two logical partitions in a channel-subsystem image, enter the following:

```
def dev 4300-430F unitadd 80 cu 8100 css 3 dasd par vm1 vm2
Device configuration change completed successfully
Ready;
```

Response 3:

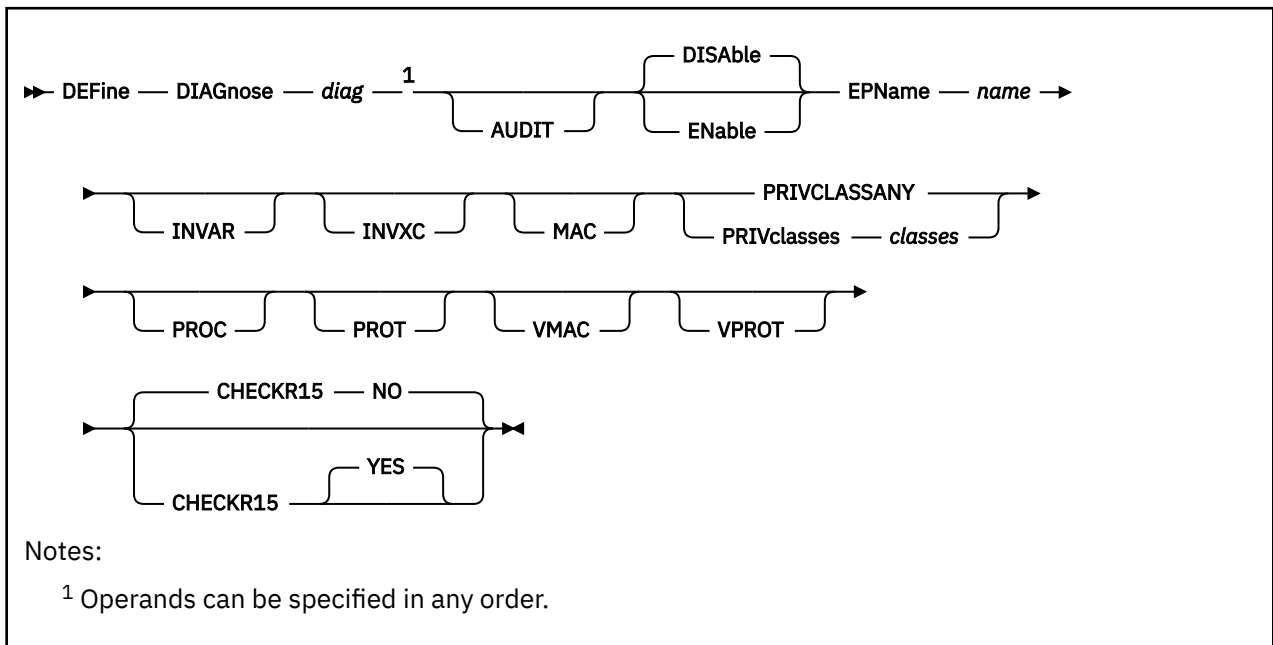
To define 16 real DASDs in two logical partitions in a channel-subsystem image and subchannel set, enter the following:

```
def dev 8180-818F unitadd 80 cu 8100 css 3 ss 1 dasd par vm1 vm2
Device configuration change completed successfully
Ready;
```

## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE DIAGNOSE



### Authorization

Privilege Class: A

### Purpose

Use DEFINE DIAGNOSE to define a new DIAGNOSE code on the system.

You can also define a new DIAGNOSE code during initialization using the DEFINE DIAGNOSE configuration file statement. For more information, see [DEFINE DIAGNOSE Statement in z/VM: CP Planning and Administration](#).

### Operands

#### *diag*

is the number of the DIAGNOSE code that you are defining. The variable *diag* is a hexadecimal number between X'0' and X'3FC' and must be a multiple of 4. The recommended variable range is a value between X'100' and X'1FC' because that is the range of diagnose codes reserved for customer use. All other DIAGNOSE code numbers are reserved for IBM use.

#### **AUDIT**

tells the external security manager (ESM) to audit the DIAGNOSE code that you are defining. When you audit a DIAGNOSE code, the ESM logs each attempt by users to issue this DIAGNOSE code.

#### **DISABLE**

tells CP not to call the entry points and external symbols associated with this DIAGNOSE code until you enable it. (For more information about enabling DIAGNOSE codes, see Usage Note “3” on page 174.) If omitted, DISABLE is the default.

#### **ENABLE**

tells CP to immediately start calling the entry points and external symbols associated with this DIAGNOSE code.

#### **EPName *name***

tells CP the name of the entry point that contains the code to process the DIAGNOSE code that you are defining. The variable *name* must be a 1- to 8-character string. The first character must be

alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

**INVAR**

tells CP not to process this DIAGNOSE code if the virtual machine issuing the DIAGNOSE code is in host access register mode.

**INVXC**

tells CP not to process this DIAGNOSE code if the virtual machine issuing the DIAGNOSE code is in Enterprise System Architecture/Extended Configuration (ESA/XC) mode or in z/XC mode.

**MAC**

tells CP to enable mandatory access control (MAC) for the DIAGNOSE code that you are defining. When MAC is enabled for your DIAGNOSE code, the external security manager (ESM) compares the security label of the virtual machine that issued your DIAGNOSE code to the security label of the resource or user that your DIAGNOSE code will affect. If you want the ESM to dynamically turn MAC on or off for this DIAGNOSE code, specify the VMAC operand.

**PRIVCLASSANY**

tells CP that users with any privilege class can issue the DIAGNOSE code that you are defining.

**PRIVclasses classes**

tells CP that only users with 1 or more of the specified privilege classes can issue the DIAGNOSE code that you are defining. The variable *classes* is 1 or more privilege classes in the range A through Z, 1 through 6, or an asterisk (\*). Privilege class \* indicates all privilege classes (A-Z and 1-6).

**Note:** If you want more than one privilege class, specify your classes in one string of characters. Do not separate the classes with blank spaces. For example, specify "privclasses abc123", not "privclasses a b c 1 2 3".

**PROC**

tells CP that, after performing the initial privilege class checks, your DIAGNOSE code processor will be responsible for any further calls to the external security manager (ESM) for the DIAGNOSE code that you are defining.

**PROT**

tells the external security manager (ESM) to protect the DIAGNOSE code that you are defining. When protection is enabled for your DIAGNOSE code, the ESM checks an access list to ensure that the user who issued your DIAGNOSE code is authorized to do so. If you want the ESM to dynamically turn protection on or off for your DIAGNOSE code, specify the VPROT operand.

**VMAC**

gives the external security manager (ESM) the power to dynamically turn mandatory access control (MAC) on or off.

**VPROT**

gives the external security manager (ESM) the power to dynamically turn DIAGNOSE code access protection on or off. If you specify PROT and VPROT for this command, you are enabling protection and allowing the external security manager (ESM) to dynamically disable that protection. If you specify VPROT and you do not specify PROT for this command, you are disabling protection and allowing the ESM to dynamically enable that protection.

**CHECKR15 YES**

indicates that the diagnose router should check register 15 upon return from the diagnose handler.

**CHECKR15 NO**

indicates that the diagnose router should not check register 15 upon return from the diagnose handler.

**Usage Notes**

1. If you do not specify the ENABLE operand, a new DIAGNOSE code is initially in a disabled state after being defined. CP treats disabled DIAGNOSE codes as if they were never defined. If you try to use a disabled DIAGNOSE code in a program, CP will give you a program check specification exception.

## DEFINE DIAGNOSE

2. To load the DIAGNOSE processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD”](#) on page 111).
3. To activate a new DIAGNOSE code after defining it, use the ENABLE DIAGNOSE command (see [“ENABLE DIAGNOSE”](#) on page 463).
4. To change the definition of an existing DIAGNOSE code after initialization, use the MODIFY DIAGNOSE command (see [“MODIFY DIAGNOSE”](#) on page 697).
5. To display information about a DIAGNOSE code (status, entry point name, and privilege class), use the QUERY DIAGNOSE command (see [“QUERY DIAGNOSE”](#) on page 885).
6. To display the address of the CP DIAGNOSE code table block for a DIAGNOSE code, use the LOCATE DGNBK command (see [“LOCATE DGNBK”](#) on page 608).
7. To deactivate a DIAGNOSE code after defining it, use the DISABLE DIAGNOSE command (see [“DISABLE DIAGNOSE”](#) on page 316).
8. Once defined, DIAGNOSE codes cannot be deleted. They may be altered in various appropriate ways, but they remain in existence until a SHUTDOWN or RESTART IPL is done.
9. To remove the DIAGNOSE processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117).
10. Many external security managers (ESMs) do not support DIAGNOSE codes above X'03FC'. For this reason, CP does not support DIAGNOSE codes above X'03FC'. The DIAGNOSE codes between X'0000' and X'03FC' are divided as follows:

**X'0000' to X'00FC'**

Reserved for IBM use

**X'0100' to X'01FC'**

Reserved for customer use

**X'0200' to X'03FC'**

Reserved for IBM use.

11. When CHECKR15 YES is specified, the diagnose router will check register 15 on return from the diagnose handler. If register 15 contains:

**RC = 0**

Processing was successful. Complete the guest instruction.

**RC = 4**

Processing failed due to a condition which would cause a guest program check. Simulate guest program interruption passed in R0.

**RC = 8**

Nullify the instruction.

**RC = 12**

Present the machine check then nullify the instruction. R2 will contain the address of the MCRBK which will contain the machine check information.

**RC = 16**

Generate machine check for processing damage, then go to HCPENDOP to terminate the instruction.

**RC = 20**

Present the machine check, then go to HCPENDOP to terminate the instruction. R2 will contain the address of the MCRBK, which contains machine check information.

**RC = 24**

Issue error message or soft abend for paging I/O error, then nullify the instruction. R1 has the message or abend number.

If a return code is invalid (negative, not a multiple of 4 or too big ( RC > 24 )), then a soft abend will occur.

12. The DEFINE COMMAND or DEFINE DIAGNOSE commands should not be used on a system with an active External Security Manager (ESM); unpredictable results could occur. To safely define a CP

command add the DEFINE COMMAND or DEFINE DIAGNOSE statement to the SYSTEM CONFIG file and re-IPL the system. This allows the command and diagnose additions to control structures to be made prior to activating the ESM.

13. For more information about user-defined DIAGNOSE codes, see [Defining and Modifying Commands and Diagnose Codes in z/VM: CP Exit Customization](#).

## Responses

Response 1:

To define a general purpose DIAGNOSE code (X'10C') that will be processed by entry point HCPSRC00, enter the following:

```
define diagnose 10c epname hcpsrc00 privclasses g
Ready;
```

Response 2:

To define DIAGNOSE code X'100' and make it available to users with privilege classes C or E, enter the following:

```
define diagnose 100 epname qwerty privclasses ce
Ready;

enable diagnose 100
Ready;

query diagnose 100
Code Status EPName PrivClasses
0100 Enabled QWERTY CE
Ready;
```

Response 3:

To define DIAGNOSE code X'18C' and make it available to all users, enter the following:

```
define diagnose 18c epname hcpsrc00 privclassany
Ready;

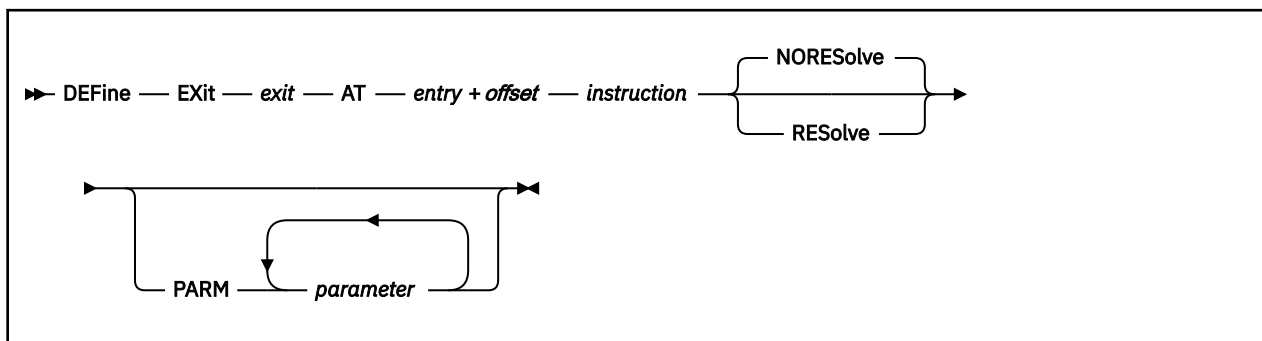
enable diagnose 18c
Ready;

query diagnose 18c
Code Status EPName PrivClasses
018C Enabled HCPSRC00 <ANY>
Ready;
```

## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE EXIT



### Authorization

Privilege Class: A

### Purpose

Use DEFINE EXIT to dynamically define a new exit point in CP.

You can also dynamically define a new CP exit point during system initialization using the DEFINE EXIT configuration file statement. For more information, see [DEFINE EXIT Statement in z/VM: CP Planning and Administration](#).

### Operands

#### *exit*

is the number of the exit point you are defining. This value must be a hexadecimal number between X'0' and X'FFFF'. The recommended value is between X'F000' and X'FFFF', because that range is reserved for customer use. All other exit numbers are reserved for IBM, vendor, or general use.

#### **AT** *entry + offset instruction*

identifies the location of the exit point you are defining and the instruction that is located at the exit point. The variable *entry* must be a 1-to-8-character string. The first character must be alphabetic or one of the following special characters: \$ (dollar sign), # (number sign), \_ (underscore), or @ (at sign). The rest of the string can be alphabetic or numeric characters, the four special characters (\$, #, \_, or @), or any combination. The variable *offset* must be a 1-to-4-character even hexadecimal number between X'0' and X'FFFE'. The variable *instruction* must be a 2-, 4-, or 6-character hexadecimal number.

#### **NORESolve**

tells CP to resolve the entry points associated with this exit number the first time they are called. This is the default.

#### **RESolve**

tells CP to resolve the entry points associated with this exit number when the association is first established. Any existing associated entry points are resolved immediately.

#### **PARM** *parameter*

is a list of one or more parameters to be supplied to the exit. Five kinds of tokens can be used to define a parameter:

1. Addresses: strings up to eight characters long, consisting of the hexadecimal digits 0 through 9 and A through F.
2. General Registers: strings beginning with G or R, followed by a decimal number between 0 and 15 or a hexadecimal digit, designating the contents of a general register.
3. Indirection: a percent sign (%), which causes the contents of an address or the contents of an address in a register to be used instead of the address or register contents itself.



4. Arithmetic: a plus sign (+) or minus sign (-).
5. Displacement: strings of up to four hexadecimal digits.

Each parameter string specifies how to combine these tokens to generate a parameter value to be passed to an exit routine. The following is a Backus-Naur definition of the syntax of a parameter:

```
<parameter> ::= <anchor> | <anchor><vector>
<anchor> ::= <reg> | 0...FFFFFFFF | <anchor>%
<vector> ::= <modifier> | <vector>% | <vector><modifier>
<modifier> ::= +<disp> | -<disp> | +<reg> | -<reg>
<reg> ::= G<digit> | R<digit>
<digit> ::= 0...15 | 0...9, A...F
<disp> ::= 0...7FFF
```

## Usage Notes

1. To load the exit processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD” on page 111](#)).
2. To assign entry points or external symbols to an exit point, use the ASSOCIATE EXIT command (see [“ASSOCIATE EXIT” on page 20](#)).
3. To activate a new exit after defining it, use the ENABLE EXITS command (see [“ENABLE EXITS” on page 465](#)).
4. To change the definition of an existing dynamic exit point, or to remove the exit point from the system, use the MODIFY EXIT command (see [“MODIFY EXIT” on page 700](#)).
5. To display information about an exit (status, entry point name, and parameters), use the QUERY EXITS command (see [“QUERY EXITS” on page 907](#)).
6. Exit numbers are allocated as follows:
  - X'0000' to X'7FFF' are reserved for IBM use.
  - X'8000' to X'FFFF' are reserved for vendor and general use.
  - X'F000' to X'FFFF' are reserved for private customer use.
7. The RESOLVE option ensures that the entry names associated with an exit point are defined.
8. Each exit is passed a parameter list that begins with three standard parameters, as described in [z/VM: CP Exit Customization](#). Additional parameters, specified by the PARM operand, are optional and follow the first three in the order in which they are specified.
9. Errors (for example, addressing exceptions) during evaluation of user-defined parameters when an exit is being invoked cause CP to abend.
10. For more information about user-defined exits, see [z/VM: CP Exit Customization](#).

## Examples

### Example 1:

To define an exit that will be entered every time a MESSAGE command is issued, enter the following:

```
define exit f422 at hcpxmgms + 4 41700000
Ready;
```

### Example 2:

To define exit X'F800', pass it two additional parameters, and associate it with entry point QWERTY, enter the following:

```
define exit f800 at hcplpg + 7ce 41204028 parm g1+8% g4
Ready;

associate exit f800 epname qwerty
Ready;

enable exits f800
Ready;
```

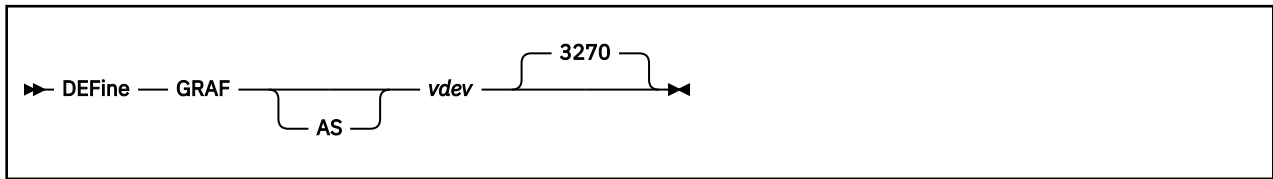
## DEFINE EXIT

```
query exits f800
Exit Status      Calls      Returns Seconds
F800 Enabled      0          0 0.000000
      Location      Instruction Resolution
      HCPLOG +0FCE 41204028  NORESOLVE
      Defined by: MAINT on 09/03/98 at 13:27:54
      PARM R1+8%
      R4
      EPNAME      Attempts      Calls Seconds
      QWERTY      0            0 0.000000
Ready;
```

## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE GRAF



### Authorization

Privilege Class: G

### Purpose

Use DEFINE GRAF to add a display station to your virtual machine configuration.

### Operands

*vdev*

**AS** *vdev*

identifies the virtual device number of the display you are defining.

**3270**

identifies the virtual device type of the display you are defining. This is the default.

### Usage Notes

1. After the display station is defined, you can enter DIAL from a real display station to logically connect the display station to your virtual machine.
2. For additional information, see [“Usage Notes” on page 122](#).

### Responses

Response 1:

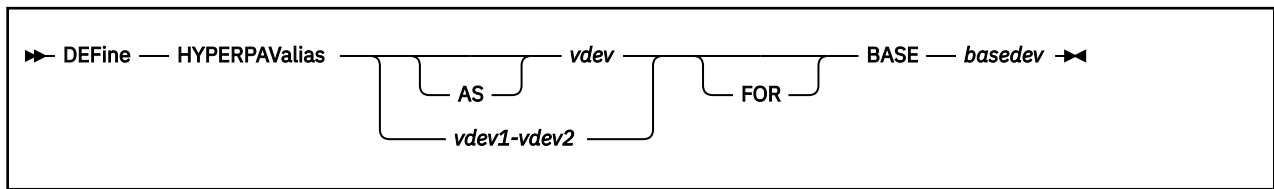
```
GRAF vdev DEFINED
```

confirms the definition of the virtual display station.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE HYPERPAVALIAS



### Authorization

Privilege Class: G

### Purpose

Use DEFINE HYPERPAVALIAS to define one or more virtual HyperParallel Access Volume (HyperPAV) aliases that will reside in the same pool as the specified base HyperPAV.

### Operands

#### AS *vdev*

#### *vdev*

is the virtual device number of the HyperPAV alias you are defining.

#### *vdev1-vdev2*

is the range of virtual device numbers of the HyperPAV aliases you are defining.

#### FOR BASE *basedev*

is the device number of an existing virtual base HyperPAV.

### Usage Notes

1. The number of virtual HyperPAV aliases cannot exceed the number of real HyperPAV aliases in the pool to which the base is assigned.
2. There can only be 254 aliases per pool, and there is a limit of 16,000 pools per system.
3. The base *vdev* specified on the command must be a virtual HyperPAV base device that is defined as a full-pack (0-END) minidisk or a 1-END minidisk.

### Responses

Response 1:

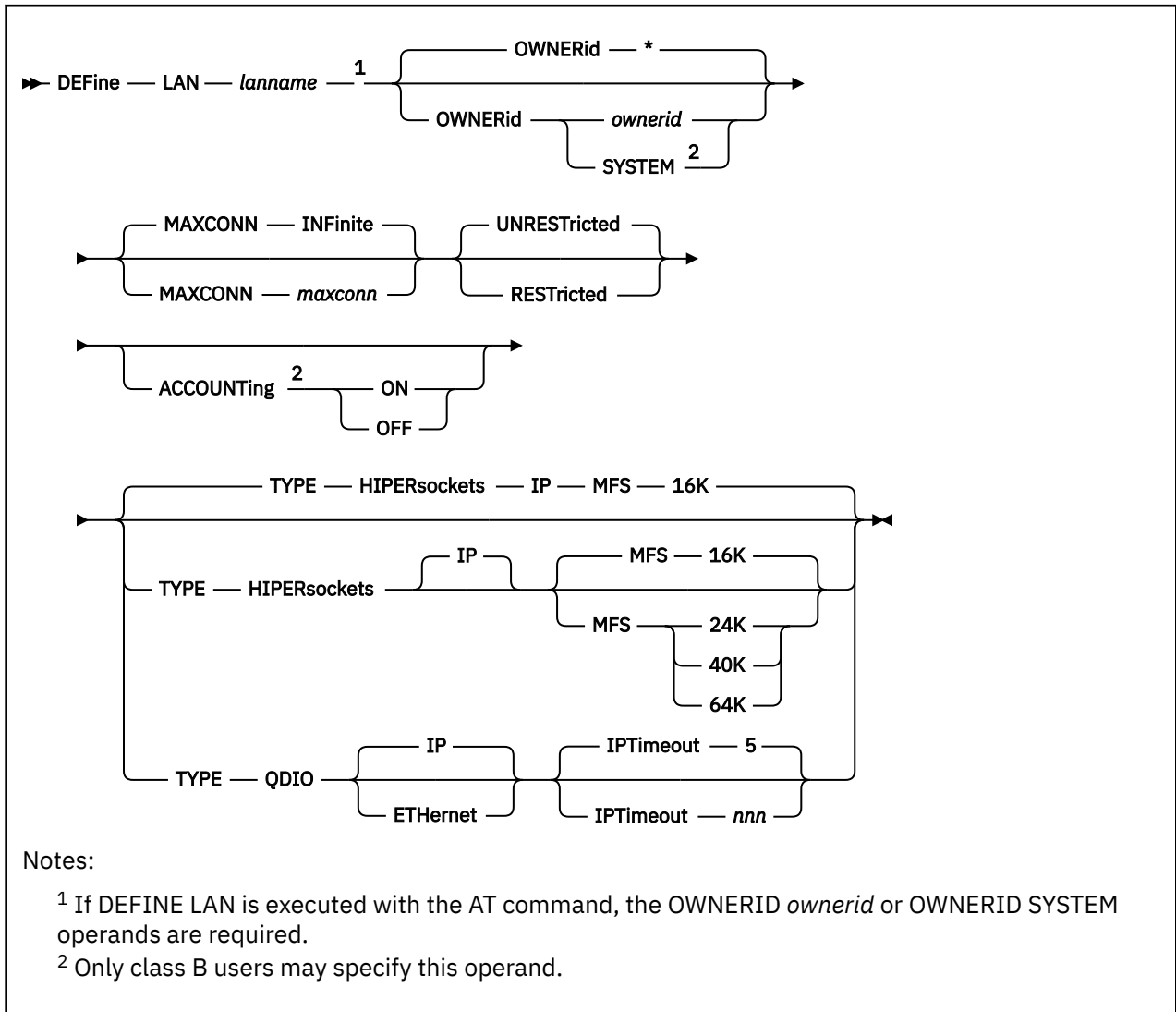
```
DASD vdev DEFINED
```

confirms that the device has been created.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE LAN



### Authorization

Privilege Class: B, G

### Purpose

Use DEFINE LAN to create a guest LAN which can be shared among virtual machines on the same VM system. Each guest LAN is identified by a unique combination of *ownerid* and *lanname*. A VM user can create a simulated network interface card (NIC) and connect it to this LAN segment.

The SET LAN command can be used to modify attributes of the guest LAN.

You can also define a LAN during system initialization using the DEFINE LAN configuration file statement.

The Class B form of the command allows the invoker to create a LAN for another user (for example, OWNERID SYSTEM) and specify whether accounting is on or off for the LAN being defined.

For more information, see [“SET LAN”](#) on page 1536, [“COUPLE”](#) on page 83, [“DEFINE NIC”](#) on page 194, [DEFINE LAN Statement in z/VM: CP Planning and Administration](#), and [Virtual Networking Options in z/VM: Connectivity](#).

## Operands

### *lanname*

is the name of the new emulated LAN (Local Area Network). The *lanname* is a single token (1-8 alphanumeric characters). The combination of *ownerid* and *lanname* will identify this LAN for subsequent commands.

### **OWNERid \***

#### **OWNERid *ownerid***

#### **OWNERid SYSTEM**

establishes the owner of the new LAN. When OWNERID \* is specified, the owner is the invoker. This is the default.

The Class G form of the command only accepts \* or the user ID of the invoker. This creates a TRANSIENT LAN.

The Class B form of the command also accepts OWNERID SYSTEM (creating a PERSISTENT LAN), or any other user ID (creating a TRANSIENT LAN).

### **MAXCONN INFinite**

#### **MAXCONN *maxconn***

sets the maximum number of simultaneous adapter connections permitted at any given time. When MAXCONN is specified as INFINITE, there is no limit on the number of connections. Any other value, *maxconn*, limits the number of simultaneous connections to a decimal number in the range of 1–1024 (inclusive).

If MAXCONN is omitted, the default is MAXCONN INFinite.

### **RESTRicted|UNRESTRicted**

sets the type of authorization required to connect to this LAN. Options are:

#### **RESTRicted**

Defines a LAN with an access list to restrict connections. The LAN owner will use the **SET LAN** command to GRANT or REVOKE access to specific VM users (by userid). The **COUPLE** command will only allow authorized users (those on the access list) to connect a simulated adapter to a RESTRICTED network.

#### **UNRESTRicted**

Defines a LAN with no access list. When CP is in control of the LAN, connections to this LAN are not restricted by user ID. All users connected to this LAN will be able to use PROMISCUOUS Mode. If an External Security Manager (ESM) is in control of the LAN, the ESM may restrict access.

When neither option is specified, the default is to define an **UNRESTRicted** LAN.

### **ACCOUNTing ON**

#### **ACCOUNTing OFF**

Allows a Class B user to control whether accounting records are created for the LAN being defined. The default setting may be changed by the VMLAN statement in the system configuration file, and queried by QUERY VMLAN.

### **TYPE HIPERsockets**

defines a guest LAN for use by simulated HiperSockets adapters. A HiperSockets LAN can only accept connections from a simulated HiperSockets adapter.

If **TYPE** is omitted, the default is **TYPE HIPERSOCKETS**.

### **MFS 16K**

### **MFS 24K**

### **MFS 40K**

### **MFS 64K**

sets the Maximum Frame Size (MFS) for adapters on this network. When an adapter is connected to this LAN, it will adopt the network MFS. The MFS value determines the amount of storage to be allocated for internal structures, and limits the effective MTU size (Maximum Transmission Unit) for the coupled adapters. For general internet communications, a lower MFS is probably better. However,

a high MFS may provide higher data transfer rates for applications that are capable of using larger packet sizes.

If MFS is omitted, the default for HiperSockets is 16K. The MFS operand is not valid for QDIO but the effective MFS is 8992 for a QDIO adapter.

### TYPE QDIO

defines a guest LAN for use by simulated QDIO adapters. A QDIO LAN can only accept connections from a simulated QDIO adapter.

### IP | Ethernet

indicates whether the transport for the LAN is ETHERNET or IP. The ETHERNET transport type is Data Link (Layer 2) based, where the ETHERNET frame is used as a point of reference for source and destination Media Access Control (MAC) addresses in transporting ETHERNET frames on the LAN. The IP transport type is Network (Layer 3) based, where the IP packet is used as the point of reference for source and destination IP addresses in transporting IP packets on the LAN. See the usage notes for more detail.

### IPTimeout *nnn*

indicates the length of time in minutes that a transient IP address table entry remains in the IP address table for the virtual network. A transient entry is an entry added to the virtual network by one guest on behalf of another. This operand has no meaning for ETHERNET networks.

*nnn* is a number from 1 to 240. 5 minutes is the default value.

## Usage Notes

1. The Class G form of the **DEFINE LAN** command only allows the invoker to create a guest LAN that belongs to the invoker. That is, *ownerid* must be the user ID of the invoker, or an asterisk (\*).
2. The Class B form of the **DEFINE LAN** command allows the invoker to create a guest LAN that belongs to another user (or to **SYSTEM**).
3. A **PERSISTENT** LAN can only be destroyed by the **DETACH LAN** command. When a guest LAN is created for ownerid SYSTEM, it is automatically classified as a **PERSISTENT** LAN.
4. A **TRANSIENT LAN** can be destroyed by the **DETACH LAN** command, or by elimination of all references. When a guest LAN is created for a user (any userid other than SYSTEM) it is automatically classified as a **TRANSIENT** LAN. During LOGOFF processing, any **TRANSIENT** LAN with no active connections will be destroyed. If active connections exist, the "orphaned" LAN will continue to be functional until the last adapter is uncoupled. However, if the owner completes LOGON before the last adapter is uncoupled, the LAN is no longer "orphaned" and it will survive the removal of the last adapter.
5. The **VMLAN** statement in SYSTEM CONFIG (or the **SET VMLAN** command) may restrict the number of guest LANs on the system. Use the **QUERY VMLAN** command to determine what limit is in effect.
6. **DEFINE LAN** statements can be added to the SYSTEM CONFIG file to automatically create guest LANs during CP system initialization. Every guest LAN defined in the SYSTEM CONFIG file is classified as a **PERSISTENT** LAN.
7. A Class B user can use the **SET LAN** and **DETACH LAN** commands to operate on a LAN segment that belongs to another user (e.g. a SYSTEM-owned LAN).
8. When a RESTRICTED LAN is defined, the owner (as determined by the DEFINE command) is automatically added to the access list unless the owner is SYSTEM. SYSTEM will not be automatically added to the access list. The owner can use the **SET LAN** command to grant and revoke access to the LAN.
9. SET LAN cannot be used to change the type of transport. The LAN will need to be redefined.
10. The TYPE QDIO guest LAN supports two modes of operation for data transport in support of both TCP/IP and non-IP based applications. In deciding which mode to deploy for your network, some things to consider about deploying an ETHERNET virtualized LAN segment are:
  - Do your servers or applications need to have their own unique MAC addresses (load balancers)?
  - Do you plan to deploy non-IP based applications on your network (SNA or NetBIOS, for example)?

## DEFINE LAN

- Do you want to build a virtual LAN segment that operates closely to its physical counterpart?

The key attributes of each transport mode with their operational characteristics are as follows:

- ETHERNET (Layer 2)
  - Supports all applications that deploy ETHERNET (IEEE 802).
  - ETHERNET frames are transported on the LAN segment.
  - All destinations are identified by MAC address.
  - MAC addresses are locally administered by the LAN administrator through z/VM CP commands or configuration statements.
  - Each host connection is identified by a single MAC address.
  - This is a Link Layer transport, in which all hosts maintain their respective ARP caches.
  - VLAN tagging resides within the ETHERNET frames per IEEE 802.1Q specifications.
- IP (Layer 3)
  - Supports IP for TCP/IP applications only.
  - IP packets are transported on the LAN segment.
  - All destinations are identified by IP addresses.
  - IP address assignments are set by the host running in the guest virtual machine.
  - Each host may have more than one IP address (multi-homed).
  - This is Link Layer independent (that is, no MAC addresses).
  - VLAN tagging resides in internal QDIO headers.

A guest LAN operates in only one mode for a given instance. For example, if the LAN is configured as IP, then all communications on the LAN segment must be IP based. The same is also true when the LAN is configured in ETHERNET mode.

### Examples

#### 1. DEFINE LAN - Class G:

To create a user LAN named QNET that will allow up to 16 connections, enter:

```
cp define lan qnet type qdio maxconn 16
LAN VMUSERX QNET is created
```

#### 2. DEFINE LAN - Class B:

To create a SYSTEM LAN named INEWS that will allow up to 100 connections and will not have accounting records created, enter:

```
cp define lan inews ownerid system maxconn 100 accounting off
LAN SYSTEM INEWS is created
```

Note that this must be done by a user with Class B privileges since the LAN is not owned by the invoker and the ACCOUNTING option is specified.

### Responses

Response 1:

```
LAN ownerid lanname is created
```

is the normal response for this command.

### General Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

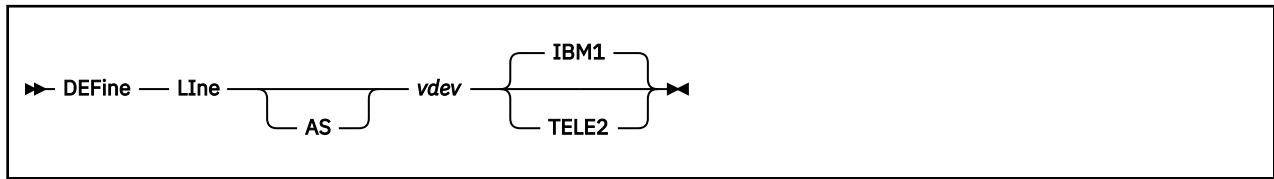


## Messages

Messages:

- HCP013E Conflicting option - *option*
- HCP2782E LAN *ownerid lanname* not created; a detach is in progress for a VSWITCH with that name
- HCP2782E LAN *ownerid lanname* not created; this name already in use
- HCP2784E LAN *ownerid lanname* not created; the system LAN limit has been reached
- HCP2784E LAN *ownerid lanname* not created; VMLAN limit has been reached
- HCP2795E You are not authorized to DEFINE LAN *lanname* for OWNERID *ownerid*
- HCP2795E You are not authorized to override LAN ACCOUNTING
- HCP2796E LAN *ownerid lanname* not created; user *ownerid* is not logged on

## DEFINE LINE



### Authorization

Privilege Class: G

### Purpose

Use DEFINE LINE to add a virtual communication line to your virtual machine configuration.

### Operands

*vdev*

**AS** *vdev*

identifies the virtual device number of the line you are defining.

**IBM1**

identifies the virtual device type of the line you are defining. This is a 2741, 3767, or equivalent device. This is the default.

**TELE2**

identifies the virtual device type of the line you are defining. This is a 3101, 3151, 3161, 3162, 3163, or equivalent device.

### Usage Notes

1. After the line is defined, you must enter DIAL from a real device to logically connect the line to your virtual machine.
2. For additional information, see [“Usage Notes” on page 122](#).

### Responses

Response 1:

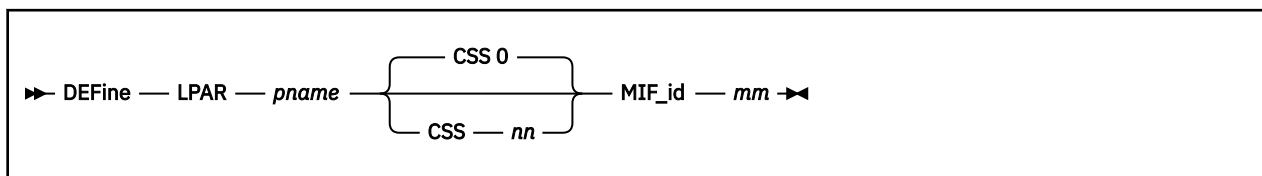
```
LINE vdev DEFINED
```

confirms the definition of the virtual communication line.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE LPAR



### Authorization

Privilege Class: B

### Purpose

Use the DEFINE LPAR command to define a logical partition to the I/O configuration.

### Operands

#### *pname*

specifies the name of the logical partition to be defined. *pname* must be a 1-character to 8-character alphanumeric string.

**Note:** The first character of *pname* must be alphabetic and the following names cannot be used: PHYSICAL, REC, SYSTEM, PRIMnnn (where nnn are decimal digits).

#### **CSS** *nn*

is the channel-subsystem ID associated with the logical partition that is being defined. The variable *nn* must be a 1-digit to 2-digit hexadecimal number between X'00' and X'0F.'

#### **MIF\_id** *mm*

is the MIF-ID within the specified CSS associated with the logical partition that is being defined. The variable *mm* must be a 1-digit to 2-digit hexadecimal number between X'01' and X'0F.'

### Usage Notes

1. The specified CSS and MIF-ID combination must be an available reserved slot in the configuration in order for a logical partition to be defined to it. An available reserved logical partition slot is created by:
  - Specifying it with the RESOURCE statement in the IOCP file.
  - Deleting a logical partition definition with the DELETE LPAR command.
2. This command requires that the logical-channel-subsystem facility and logical-partition-configuration facility be installed on your system.

### Examples

To define logical partition VMSYSONE to CSS 1 with a MIF-ID of 3, enter the following:

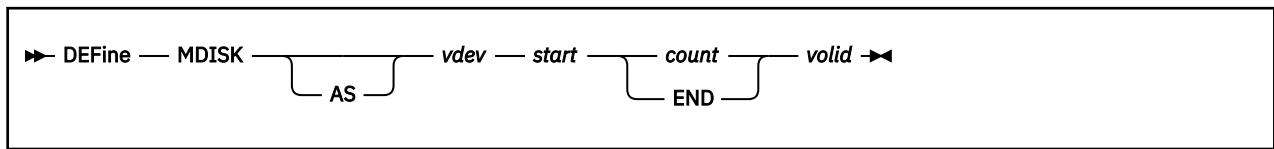
```

define lpar vmsysone css 1 mif 3
Logical partition configuration change completed successfully
Ready;
  
```

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE MDISK



### Authorization

Privilege Class: A

The user must be the primary system operator or the user's OPTION directory statement must include the DEVMAINT option.

**Note:** When the command is controlled by an external security manager (ESM), such as RACF, the ESM will check the authorization. Only privilege class A is required. No additional checks for the primary system operator or the user's OPTION directory statement are done.

### Purpose

Use DEFINE MDISK to define a minidisk on the following media:

- A system volume
- The DASD volume that is attached to the system

### Operands

*vdev*

**AS** *vdev*

identifies the virtual device number of the minidisk you are defining.

**start**

is a decimal number that defines the cylinder (CKD/ECKD) or block (FBA) on the real DASD that is cylinder 0 or block 0 for this minidisk.

Generally, ECKD user minidisks must begin with cylinder 1 or higher and FBA user minidisks must begin with block 32 or higher. If the SALIPL utility was used to install a copy of the Stand-Alone Program Loader (SAPL), then FBA user minidisks must begin with block 208 or higher.

**count**

**END**

defines the size of the minidisk.

The variable *count* is a decimal number that defines the number of cylinders or blocks to make up the minidisk.

END defines the minidisk to extend to the end of the real DASD.

**valid**

identifies the volume serial number (1- to 6-alphanumeric characters long) of the real DASD volume against which the MDISK is being defined.

### Usage Notes

1. The minidisk that you define remains available until you log off or detach the virtual device on which it resides. Use this command only for disaster recovery because it provides access to any area of a DASD volume without requiring more authorization.
2. You can define a minidisk as a full-pack minidisk. You must define the starting cylinder number as zero and include all DASD volume primary cylinders and some or all of its alternate cylinders.

3. Minidisks that are created by using the DEFINE MDISK command have cache access if they are on a cached control unit.
4. Minidisks must not be defined on PAGE, SPOL, TDSK, or on DRCT allocation areas.
5. FBA minidisks intended to be eligible for minidisk mapping and minidisk caching must start and end on a 4 K boundary. They must be defined with a number of blocks that comprise a complete number of pages (a multiple of 8 blocks) because partial pages are not formatted by ICKDSF CPVOLUME.
6. A minidisk, including a temporary disk, is not eligible for minidisk cache if it has been defined with greater than 32767 cylinders. This does not apply to FBA devices. It applies only to ECKD devices.
7. If you define a minidisk as virtual device number 192, the following special rules apply when you IPL CMS:
  - If 192 is unformatted, CMS formats it and accesses it as file mode D.
  - If 192 is CP-formatted, CMS reformats it for CMS use and accesses it as file mode D.
  - If 192 is CMS-formatted and accessed as a file mode other than D, CMS reaccesses it as file mode D.

When CMS accesses a 192 minidisk as file mode D, any minidisk or SFS directory that is already accessed as D is released.
8. When linking to a minidisk that has at least one cylinder that is the target of a persistent FlashCopy relationship, warning message HCP116I is issued. The link is not prevented when a FlashCopy relationship exists.
9. Minidisk cache is not supported for a minidisk that is defined on an Extended Address Volume (EAV) DASD, such as the 3390-0E, because of the large number of cylinders that are supported by that device.
10. For additional information, see [“Usage Notes” on page 122](#).

## Responses

Response 1:

```
DASD vdev DEFINED
```

confirms the definition of the DASD.

### **vdev**

identifies the virtual device number of the minidisk you define.

Response 2:

When minidisks that are in a FlashCopy relationship are defined, you see something like this:

```
DEFINE MDISK 200 100 100 PACK01
HCPLNS116I WARNING: Cylinders 50 through 59 of minidisk 0200 are the
HCPLNS116I          target of FlashCopy relationship 4AF7DB61.
Ready; T=0.01/0.01 14:48:25

DEFINE MDISK 300 100 100 PACK02
HCPLNS116I WARNING: Cylinders 50 through 59 of minidisk 0300 are the
HCPLNS116I          target of FlashCopy relationship 4AF7BDB3.
Ready; 0.01/0.01 14:48:29
```

## Messages

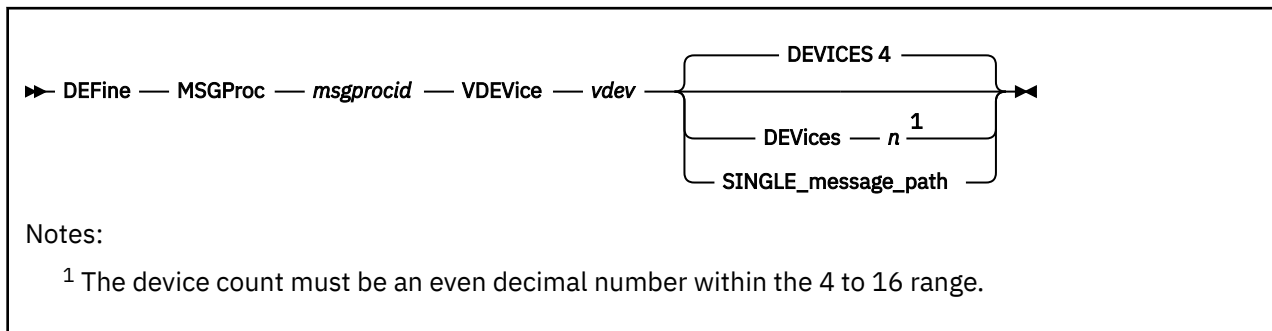
- HCP023E VOLID missing or invalid
- HCP024E {CYLINDERS|PAGES} MISSING OR INVALID {Block number|Cylinder|Cylinder or block numbers} missing or invalid
- HCP101E DASD vdev forced R/O; R/O by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}]; stable by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}]

## DEFINE MDISK

- HCP102E DASD *vdev* forced R/O; R/W by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; stable by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}}
- HCP103E DASD *vdev* forced R/O; R/W by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}, R/O by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; stable by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}}
- HCP104E *userid vdev* not linked; R/O by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname* }; stable by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; exclusive by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}}
- HCP105E *userid vdev* not linked; R/W by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; stable by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}; exclusive by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}}
- HCP106E *userid vdev* not linked; R/W by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}, R/O by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname* }; stable by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}}
- HCP107E *userid vdev* not linked; not in CP directory
- HCP108E *userid vdev* not linked; valid *valid* not mounted
- HCP109E *userid vdev* not linked; invalid link device
- HCP110E *userid vdev* not linked; *type vdev* already defined
- HCP115E {LINK|AUTOLOG|XAUTOLOG} invalid; excessive incorrect passwords
- HCP117E Userid *userid* not linked; valid *valid* conflict
- HCP153E Device {*vdev|range*} exceeds maximum number of virtual devices
- HCP1150E DASD {*rdev|vdev*} is not a valid base exposure.
- HCP1151E *userid vdev* has not been linked because it is not within CP volume cylinder extents.
- HCP1152E *userid vdev* has not been linked because it would overlap system {*directory|spool|paging|temporary disk*} space
- HCP2880E *userid vaddr* is not linked, CSE LINK I/O error, code *cc*.

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE MSGPROC



### Authorization

Privilege Class: G

### Purpose

Use the DEFINE MSGPROC command to define a virtual message processor and associated message devices in the issuer's virtual I/O configuration. This command establishes a coupling connection between a CFUSER defined user ID and the specified CF Service Machine running under z/VM.

### Operands

#### *msgprocid*

specifies the user ID of a z/VM Virtual Machine that is defined to run as a Coupling Facility. CP will establish a coupling connection between the user ID that issues this command and the user ID specified by *msgprocid*.

#### VDEvice *vdev*

is the virtual device number of the base message device that you are defining to CP. The device number must be a 1 to 4 digit hexadecimal number between X'0000' and X'FFFC'. The virtual device number specified will be the first of four consecutive device numbers created. If the specified device number is not valid or if it or any of the next three consecutive device numbers are already defined, this command will fail.

#### DEVices *n*

specifies the number of message devices to be created in the coupling guest. The number of devices (*n*) must be an even number between 4 and 16 inclusive; the default is 4.

#### SINGLE\_message\_path

will force only a single message path to be created for each message device. By default this command will create two message paths per message subchannel. Specifying SINGLE will turn off multi-path simulation when communicating with the Message Processor being defined.

### Usage Notes

1. The user must be authorized to issue this command via the CFUSER operand on the OPTION directory statement in the user directory. Without authorization, the MSGProc operand is rejected as not valid.
2. The SPECIAL directory statement allows the system administrator to restrict which message processors that a user is allowed to define. If SPECIAL directory statements are used to define the message processors for the user, then the user can specify only those *msgprocid*s on the DEFINE MSGPROC command. Specifying a *msgprocid* not defined by the SPECIAL directory entry causes the operand to be rejected as not valid.

## DEFINE MSGPROC

3. The *msgprocid* must be set up and authorized to run as a CF Service Machine via the CFVM operand on the OPTION directory statement in the user directory.
4. The CF Service Machine that you specified must be active and running prior to issuing this command. If the CF Service Machine is not running disconnected, no message devices will be defined.
5. A single coupled guest can connect to a maximum of 8 different CF Service Machines at the same time.
6. The *vdev* chosen must be a hexadecimal number low enough for the addition of *vdev* plus the number of DEVICES minus 1 with a result less than or equal to X'FFFF'.

You can only have one connection with each CF SVM. Therefore, if you want to change the *vdev* number from one you have already defined, you need to use the DETACH MSGPROC command to first detach the virtual message processor and then DEFINE MSGPROC command to define a new message processor.

7. When DEVICES *n* is used, specify between 4 and 16 inclusive; this device count must be even. If DEVICES is not specified, the default is 4.

You can have only one connection with each CF SVM. Therefore, if you already have defined a device using the default 4 and want to define a different device number, you need to use the DETACH MSGPROC command to first detach the virtual message processor and then use DEFINE MSGPROC command to define a new message processor.

8. If *msgprocid* is not z/Architecture mode, CP forces the DEVICES value to 4 no matter what was specified. The only indication of this is when confirmation message HCP2804I is displayed containing the actual device count that was used.
9. To display information about virtual message processors in your I/O configuration, use the QUERY VIRTUAL MSGPROC command.
10. To display information about the virtual message devices within your I/O configuration, use the QUERY VIRTUAL MSGDEVICES command.
11. To restart a virtual message processor that is within your I/O configuration, use the RESTART MSGPROC command.
12. To remove all message facility capability from your virtual machine, use the SET MSGFACIL OFF command.
13. To detach a virtual message processor and associated virtual message devices, use the DETACH MSGPROC command.
14. CHPIDs 8X and CX are used for the message devices. 'X' depends on when the connection was established between the coupling guest and the coupling facility relative to when other connections were made. A conflict exists when an OSA device is attached to a CFVM user and the OSA's CHPID is the same as that of a message device associated with the CFVM. When this happens, the following MVS errors are received when trying to bring the OSA online:

```
IEE103I UNIT 0420      NOT BROUGHT ONLINE
IEE763I NAME= IOSVDSE0 CODE= 0000000800000000
IOS576I OSA DEVICES REQUIRE AN OSA CHANNEL PATH BUT
TYPE 0B FOUND TYPE=COUPLING FACILITY SENDER
```

15. For more information, see the DEFINE command.

## Responses

Response 1:

To define a connection with four message devices and two Message Paths to a CF Service Machine user ID CFCC1, enter the following:

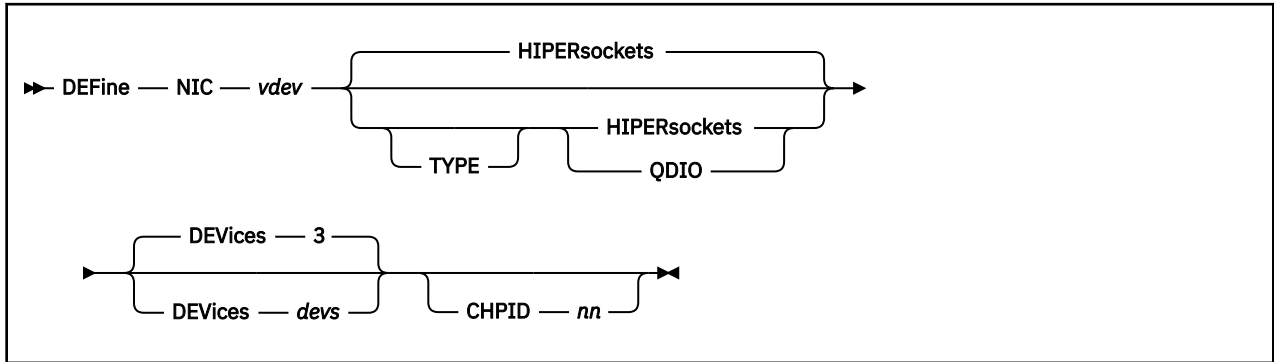
```
def msgp cfcc1 vdev 700
HCP2804I Message devices 0700-0703 defined and coupled to CFCC1.
Ready;
```



## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE NIC



### Authorization

Privilege Class: G

### Purpose

Use the DEFINE NIC command to install a simulated network interface card (NIC) in the invoker's virtual machine configuration. A successful definition creates a contiguous range of virtual OSA-type devices which all belong to the same simulated adapter. After the network interface card (NIC) is installed, use the **COUPLE** command to connect the adapter to a guest LAN or virtual switch. VM guests connected to the same LAN segment or virtual switch can exchange messages using the same communication software that they would use to drive a physical network adapter.

For more information on the COUPLE command, see [“COUPLE” on page 83](#).

### Operands

#### *vdev*

specifies the base virtual device address for the new adapter. The network interface card (NIC) is represented by a series of I/O devices in your virtual machine configuration. CP messages and responses will use this first device address, the base device address, to refer to the adapter.

#### TYPE

specifies the type of NIC adapter to be created, specifically the hardware and protocol that the adapter is to emulate.

#### **HIPERsockets**

defines this adapter as a simulated HiperSockets NIC. This adapter will function like the HiperSockets internal adapter (device model 1732-05). A HiperSockets NIC can function without a guest LAN connection, or it can be coupled to a HiperSockets guest LAN.

You will receive an error if you attempt to connect a simulated HiperSockets adapter to a virtual switch.

#### **QDIO**

defines this adapter as a simulated QDIO NIC. This adapter will function like OSA-Express (QDIO) adapter (device model 1732-01). A QDIO NIC is functional only when it is coupled to a QDIO guest LAN or a QDIO virtual switch.

#### **DEVICES 3**

#### **DEVICES *devs***

determines the number of virtual devices associated with this adapter. For a simulated HiperSockets adapter, *devs* must be a decimal value between 3 and 3,072 (inclusive). For a simulated QDIO adapter, *devs* must be a decimal value between 3 and 240 (inclusive). The **DEFINE NIC** command will create a range of virtual devices from *vdev* through *vdev+devs-1* to represent this adapter in your virtual machine configuration.

If the **DEVices** operand is omitted, the default is **3** for any adapter type.

### CHPID *nn*

is the two-digit hexadecimal number that represents the CHPID number the invoker wants to allocate for this simulated adapter. If the requested CHPID number is available, all of the virtual devices belonging to this adapter will share the same CHPID number.

### TYPE

This is an optional keyword that you can specify with HIPERsockets or QDIO.

## Usage Notes

1. The CP User Directory can be used to automatically define a simulated network interface card (NIC) during LOGON processing. The **NICDEF** statement supports device type **HIPERs** (to create a simulated HiperSockets adapter) or **QDIO** (to create a simulated QDIO adapter).
2. The HiperSockets adapter is functional immediately after it is defined, but it can only communicate with its own internal LAN until it is connected to a guest LAN by the COUPLE command. If an attempt is made to connect to a guest LAN via the NICDEF statement or the COUPLE command, the default internal LAN is destroyed and the adapter becomes "LAN NOTREADY" whenever it is uncoupled.
3. The virtual devices created by the DEFINE NIC command cannot be moved or detached by virtual device number. Each device will retain its assigned virtual device number until the virtual adapter is destroyed by the DETACH NIC command (or by LOGOFF processing).
4. The **CHPID** *xx* operand allows the invoker to request a specific CHPID number for this adapter. If the designated real CHPID number *xx* is available on the LPAR, it will be allocated for the adapter, and assigned to each virtual device that is created as part of this virtual adapter.

This option is useful if you need to configure a virtual environment with predictable CHPID numbers for your simulated adapters. It is also useful if you need to carry certain attributes associated with the real CHPID to the virtual adapter. An example would be the NETID associated with an OSD CHPID used as a VSWITCH uplink port. These attributes would be picked up by the internal discovery code run by the guest operating system using the virtual adapter.

See ["Channel path usage" on page 195](#) for more details about how this command interacts with real CHPIDs on the LPAR and how the CHPID option can make that interaction more predictable.

5. z/VM supports virtual QDIO networking connections comprised of one read control device, one write control device, and up to eight data devices. This provides the ability to configure up to ten virtual devices per host QDIO connection.

## Channel path usage

It is important to have a good understanding of how a Channel Path ID (CHPID) is associated with the set of simulated NIC devices created with a single invocation of this command. A CHPID is a required element for the virtual devices created by this command. If the CHPID operand is not specified on this command, VM will automatically assign a CHPID. Except for guests defined with the OPTION CHPIDV user directory statement, a VM guest's set of CHPIDs is directly related to the VM LPAR's real set of CHPIDs. That is, the VM LPAR's real set of CHPIDs pass-through to the guest.

Since the devices created by this command have no real device counterpart, VM uses an algorithm to assign a CHPID to these virtual devices that can coexist with VM's *real* set of CHPIDs. This algorithm first looks for a CHPID that is unassigned in both VM's *real* set of CHPIDs and the guest's set of CHPIDs. This lookup starts at *real* CHPID 0 and increases to 255 (xFF) until a match is found. If an unassigned CHPID is **not** found, a new lookup will start attempting to find a *real*, assigned CHPID that fulfills these requirements:

1. It has the same type needed by the devices created by this command.
2. It is a CHPID not already assigned to the guest.

This lookup also starts at real CHPID 0 and increases to 255 (xFF) until a match is found. The DEFINE NIC command will fail if a match is **not** found.

This algorithm usually works seamlessly unless a dynamic I/O event occurs that impacts the *real* CHPID picked up and assigned to devices created by this command. For example, if *real* CHPID 0 is assigned to guest virtual NICs 600-602, and a later dynamic I/O event adds *real* CHPID 0 with a non-compatible type, like a FICON CHPID, the guest using these virtual devices will likely start to experience error conditions. The errors likely will be experienced later when the guest operating system (OS) executes self-discovery code and notices the incompatible CHPID type. For example, a previously-working, second-level VM guest might start to encounter VSWITCH initialization errors on a re-IPL.

If a guest experiences a CHPID conflict due to a dynamic I/O change, one can detach the virtual devices with the conflicting CHPID, using the DETACH NIC command, and issue a DEFINE NIC command again to let VM locate a new *real* CHPID to associate with the virtual devices. This will likely also require a guest reboot or recycle of the guest's network interface.

The best way to avoid these CHPID conflicts is to keep a free set of CHPIDs available that can be used with the CHPID operand on each invocation of this command. The rule is that each invocation of DEFINE NIC must have a new available CHPID for the guest; note, however, that the same *real* CHPIDs can be used across different guests. The privileged Q CHPIDS command will display the *real* CHPID layout used by a VM LPAR. The guest OS will likely also have a means for determining its CHPID layout. For example, a second-level VM system can use the same Q CHPIDS command second-level.

If it is not possible to keep a free set of *real* CHPIDs available, remember that assigned CHPIDs can also be used if they are a compatible type. For example, an "Open Systems Adapter Direct Express" CHPID can be used for a DEFINE NIC with the TYPE QDIO operands. An "Internal Queued Direct Communications" (IQDC) CHPID can be used for a DEFINE NIC with the default TYPE HIPERSOCKETS operands. The type associated with a *real* CHPID can be determined using the privileged Q CHPID command with TYPE operand.

It should be noted that use of the OPTION CHPIDV user directory statement for a guest will avoid the pass-through of the real CHPID set to the guest. However, although use of CHPIDV completely virtualizes a guest's CHPID set, it should only be specified when a guest is going to be relocated and should not be specified for a z/OS guest.

### Examples

Create a simulated HiperSockets adapter with (3) devices.

```
cp define nic 500 hipersockets devices 3
NIC 0500 is created; devices 0500-0502 defined
```

```
cp define nic 5120 qdio devices 16
NIC 5120 is created; devices 5120-512F defined
```

### Responses

Response 1:

```
NIC vdev is created; devices vdev-lastdev defined
```

#### *vdev*

is the base device address

#### *lastdev*

is the last device address in the range created for this adapter ( $lastdev = vdev + devs - 1$ ).

This is the normal response for this command.

### Messages

For the list of general messages for the DEFINE command, see ["Messages" on page 122](#).

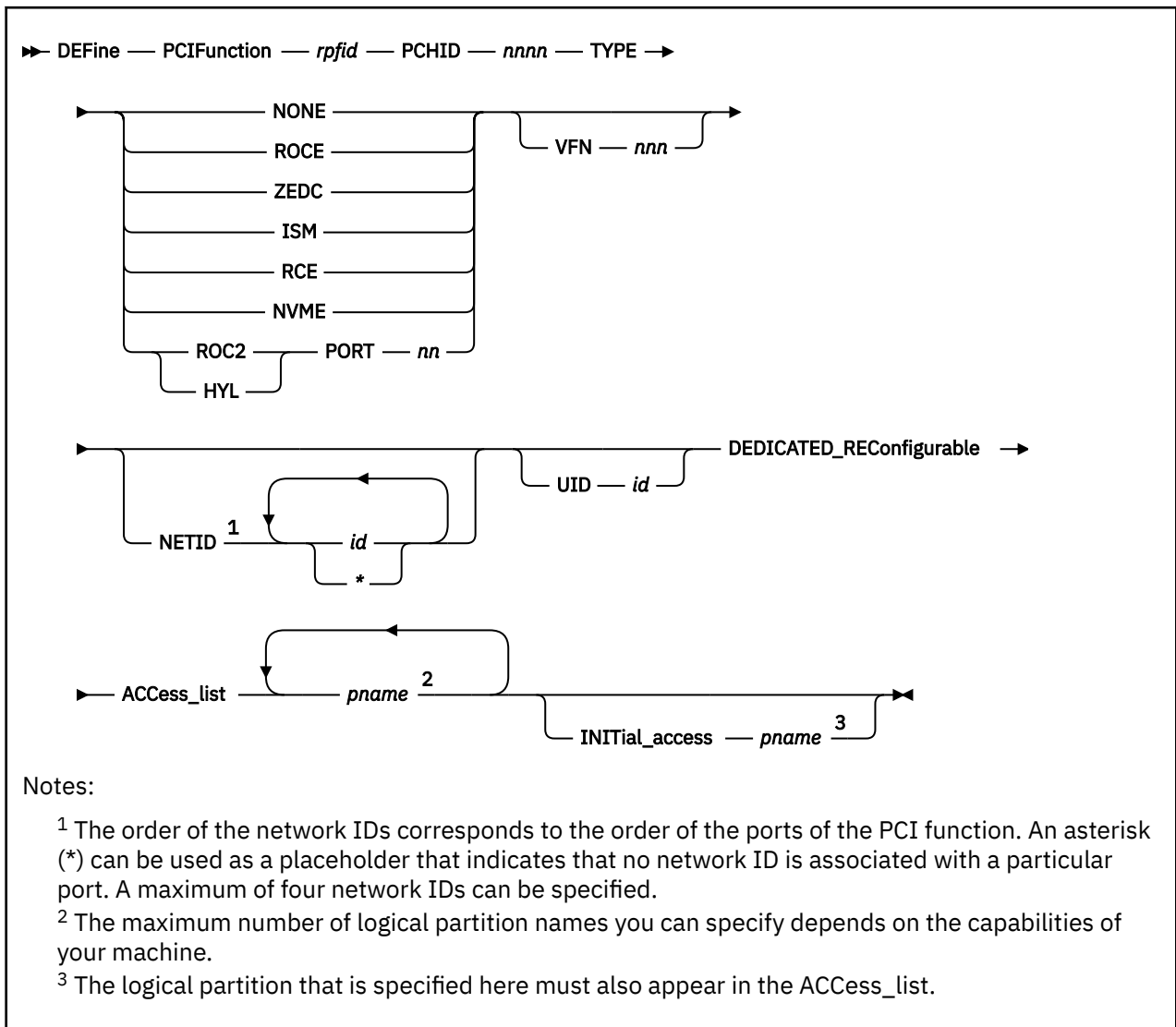
Messages:

- HCP013E Conflicting option - *option*

- HCP2794E NIC *vdev* not created; *devs* is outside the valid range of *mindevs* to *maxdevs*
- HCP2797E NIC *vdev* not created; CHPID *nn* is already in use
- HCP2797E NIC *vdev* not created; MACID *macid* is already in use [by *user ID*]
- HCP2797E NIC *vdev* not created; Single System Image mode not stable.
- HCP2797E NIC *vdev* not created; Guest relocation active.
- HCP2797E NIC *vdev* not created; Unsupported type.



# DEFINE PCIFUNCTION



## Authorization

Privilege Class: B

## Purpose

Use DEFINE PCIFUNCTION to define a real PCI function to the I/O configuration.

## Operands

### *rpfid*

is the real function identifier of the PCI function you are defining to the I/O configuration. The *rpfid* must be a 1- to 8-digit hexadecimal number between X'00000000' and X'7FFFFFFF'. The maximum value of the *rpfid* is model-dependent and your machine likely supports a lower maximum value.

## DEFINE PCIFUNCTION

### **PCHID *nnnn***

is the physical channel ID associated with the PCI function you are defining. The variable *nnnn* must be a 1- to 4-digit hexadecimal number between 'X'0000' and 'X'FFFF'. Your machine likely supports a lower maximum value. Only PCHIDs that are physically installed on your machine should be entered.

**Note:** When TYPE ISM is specified, the PCHID operand is a virtual channel ID and must be in the range of 'X'7C0' to 'X'7FF'.

### **TYPE NONE**

### **TYPE ROCE**

### **TYPE ZEDC**

### **TYPE ISM**

### **TYPE ROC2**

### **TYPE RCE**

### **TYPE HYL**

### **TYPE NVME**

indicates the type of PCI function that is being defined. The NONE operand is required on systems that do not support the TYPE operand. The ROC2 operand is used to define RoCE Express 2 and RoCE Express 3 PCI functions.

### **PORT *nn***

is the physical port associated with the PCI function you are defining. The variable *nn* must be a 1- to 2-digit decimal number in the range 1 - 15.

### **VFN *nnn***

is the virtual function number associated with the PCI function you are defining. The variable *nnn* must be a 1- to 3-digit decimal number in the range 1 - 256. Your machine likely supports a lower maximum value.

**Note:** If a virtual function number is specified for a PCHID, all definitions for that PCHID must specify a virtual function number. Some systems require the VFN operand to be specified and reject a request with return code 010D if it is not specified.

### **NETID *id***

### **NETID \***

is the network identifier associated with a port of the PCI function. A maximum of four network IDs can be specified. The order of the network IDs corresponds to the order of the ports of the PCI function. An asterisk (\*) can be used as a placeholder that indicates that no network ID is associated with a particular port. Each *id* must be a 1- to 16-digit character string.

### **UID *id***

is the user-defined identifier to be associated with this PCI function and other PCI functions that are deemed equivalent by the user. Each *id* must be an integer in the range 1 - 65535 . This field is model-dependent and the command fails if the system does not support the UID for real PCI functions.

### **DEDICATED\_REConfigurable**

tells CP that this is an unshared PCI function which can only be configured to one of the logical partitions specified on the ACCESS\_LIST operand at a time.

### **Access\_list *pname***

specifies one or more names of the logical partition(s) in which the PCI function can be configured. The maximum number of logical partition names you can specify depends on the capabilities of your machine. Each *pname* must be a 1- to 8-digit character alphanumeric string.

### **INITial\_access *pname***

specifies the name of the logical partition that has initial access to this PCI function.

**Note:** When a logical partition is specified with the INITial\_access parameter, it must also be specified as part of the ACCESS\_list.

## Usage Notes

1. Specifying Logical Partition Names:



- a. When you specify DEDICATED\_RECONFIGURABLE you must specify at least one logical partition name for ACCESS\_LIST, and you can optionally choose to specify a logical partition name for INITIAL\_ACCESS. However, after the command runs successfully, the PCI function appears in standby state (offline) to all active partitions that are specified in the ACCESS LIST. The PCI function appears in standby state whether a partition is specified on the INITIAL ACCESS list or not. How many logical partition names you can specify depends upon the capabilities of your machine; see [Table 16 on page 201](#).

Table 16. Maximum Number of Logical Partition Names		
Operand	ACCESS_LIST	INITIAL_ACCESS
DEDICATED_RECONFIGURABLE	MAX	1

2. After successfully entering this command, the software's view of the I/O configuration is changed. The hardware's view of the active I/O configuration in the hardware system area (HSA) is also changed. You must also change the hardware's view of the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must complete the following steps:

- a. Update the IOCP source file.
  - b. Run the IOCP utility against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
  - c. Issue the SET IOCDS\_ACTIVE command to make the new IOCDS file the active file that the processor uses during the next power-on reset (POR).
3. For more information about defining real PCI functions in your I/O configuration, see [z/VM: I/O Configuration](#).
  4. To change the definition of an existing PCI function on the system, use the MODIFY PCIFUNCTION command.
  5. To delete an existing PCI function on the system, use the DELETE PCIFUNCTION command.
  6. To display the status of one or more real PCI functions, use the QUERY PCIFUNCTION command.

**Note:** The QUERY PCIFUNCTION command displays the status of the PCI function only if you issue the QUERY command in a logical partition (LP) that has access to the PCI function.

7. All virtual functions for a PCHID that specify NETID values must use the same values for each port of the function. The DEFINE command fails with hex return code 0320 if the specified NETIDs do not match the existing settings.

## Responses

Response 1:

The following command and response shows a successful definition of a dedicated-reconfigurable PCI function with three logical partitions.

```
def pcif 43 pchid 8181 dedicated_rec access vm1 vm3 vm5 init vm1
PCI function configuration change completed successfully
Ready;
```

## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

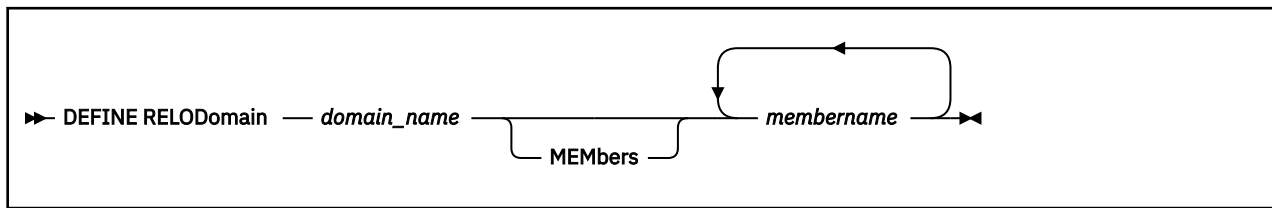
Messages:

- HCP1006E Command processing cannot complete because the system does not support the Dynamic-I/O-for-I/O-adapters facility.

## DEFINE PCIFUNCTION

- HCP1006E Command processing cannot complete because the system does not support the Utility-String facility.
- HCP6806E Dynamic I/O change request for PCI function failed with return code *nnnn*
- HCP6873E PCI function *rpfid* has been created but is in definition error and has not been added to the configuration.

## DEFINE RELODOMAIN



### Authorization

Privilege Class: A

### Purpose

Use to define or dynamically alter a named subset of the SSI membership known as a relocation domain. The domain specifies the set of systems among which a guest may relocate. The domain imposes a common set of architectural characteristics that ensures a guest is presented with a consistent architecture whatever member of the domain the guest runs in. When a user first logs on, its default associated domain is the entire SSI. (An exception is that the domain for a multi-configuration virtual machine is the singleton domain of its execution member.) A VMRELOCATE statement may be coded in the user's directory entry to assign the user to a domain other than the default at login time.

### Operands

#### **domain\_name**

The eight-character name of the domain. The reserved word "SSI" is excluded from use because it is predefined as the domain which includes all members of the SSI. Also, the names of SSI member systems are predefined as singleton domains and may not be modified by this command. The name may consist of up to eight characters using only A-Z and 0-9.

#### **MEMbers membername**

A list of one or more SSI member system names comprising the domain.

### Usage Notes

1. The QUERY SSI command may be used to determine the list of members in the SSI which can be included as members in a relocation domain.
2. The QUERY RELODOMAIN command may be used to determine the list of members in a particular relocation domain.
3. Changing the member list of an existing domain might result in changing the virtual architecture level (VAL) of that domain. If additional features are added to the VAL (for example, if the only member of the domain on an older machine type is removed, leaving all newer machine types), then guests assigned to this domain may begin to use the added features (that is, features available only on the newer machine types). Conversely, if the change results in removal of features from the VAL (such as adding an older machine type into the domain), this removal will not take effect for an active guest until its next IPL. In the interim, the guest will continue to have access to the former feature set; however, it will be excluded from relocating to any newly added member system (on the older machine type) that lacks those features.
4. For information about [Using Relocation Domains](#), see *z/VM: CP Planning and Administration*.

### Responses

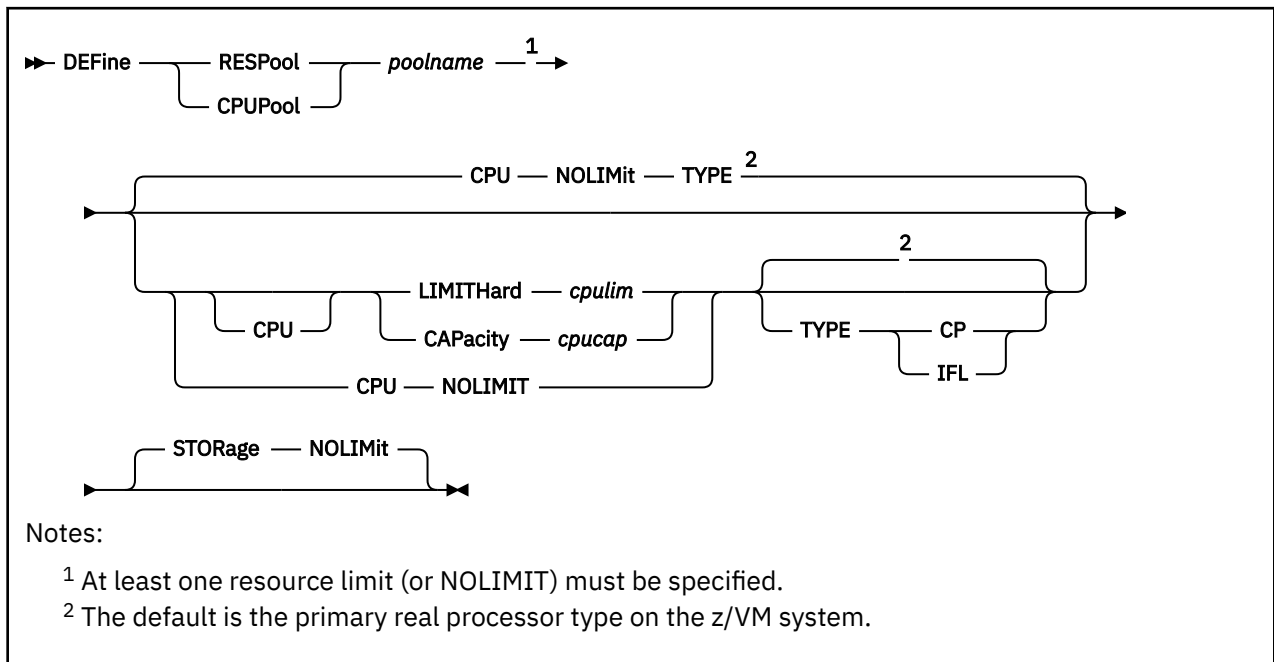
Responses issued by DEFINE RELODOMAIN

This system is not a member of an SSI cluster.

### Messages

- HCP0002E Invalid operand - *operand*
- HCP0003E Invalid option - statement contains extra options starting with *option*
- HCP0013E Conflicting option - *option*
- HCP0026E Operand missing or invalid
- HCP1814E *domain\_name* is not a valid relocation domain name
- HCP1815E Relocation domain *domain* could not be defined because *system\_name* is not an SSI cluster member
- HCP1817E The DEFINE RELODOMAIN command cannot be processed at this time
- HCP1819E The command could not be processed because the relocation domain specified is marked for deletion
- HCP1827E The maximum number of relocation domains has already been defined
- HCP6706E Invalid string - *string*
- HCP6709E Too many items specified - *token*

## DEFINE RESPOOL



### Authorization

Privilege Class: A

### Purpose

Use DEFINE RESPOOL to add a new resource pool to your system configuration. After a pool has been created, it remains until the next time z/VM is IPLed or the DELETE RESPOOL command is issued to remove it. For compatibility with prior releases, CPUPOOL is accepted as a synonym for RESPOOL.

### Operands

#### *poolname*

is the 1- to 8-character name of the pool to be created. The pool name is a single alphanumeric token that identifies a pool of resources that can be associated with guests who are to be limited.

#### CPU LIMITHard *cpulim*

#### LIMITHard *cpulim*

sets the enforcement of hard limiting for the defined pool such that the pool does not receive more than *cpulim* percent of the shared logical cores of the type specified on the TYPE operand. Here, "logical core" refers to a logical CPU when multithreading is not enabled, or to a set of logical processors (threads) currently dispatched on a physical processor core when multithreading is enabled. The *cpulim* value is a decimal whole number (no decimal place) from 1 to 100, followed by a percent sign (for example, 2% or 80%).

#### CPU CAPacity *cpucap*

#### CAPacity *cpucap*

sets the enforcement of CPU capacity limiting for the defined pool such that it does not receive an amount of processor power greater than the equivalent of *cpucap* real cores of the type specified on the TYPE operand (although the number of logical cores of that type in use simultaneously by the group can exceed *cpucap*). The *cpucap* value is a decimal number (up to two decimal places) from 0.01 to 999 (for example, 2.5 or 8).

## DEFINE RESPOOL

### TYPE

specifies the type of virtual CPU being limited. If no type is specified when a resource pool is defined, it defaults to the primary real processor type on the z/VM system. The valid CPU types are:

### CP

indicates a central processor.

### IFL

indicates an IBM Integrated Facility for Linux processor.

### CPU NOLIMit

prevents the enforcement of CPU capacity limiting for the defined pool.

### STORAge NOLIMit

prevents the enforcement of memory limiting for the defined pool.

## Usage Notes

1. The DEFINE RESPOOL command can limit only CP and IFL CPU types and is intended to create a pool for groups of guests that use only these types of CPUs, such as a group of Linux virtual machines that use only IFLs.
2. CPU limiting can work only when CONSUMPTION limiting is in effect on the z/VM system. This setting (SET SRM LIMITHARD CONSUMPTION) is the default. As a result, a pool can be created only if the system is set to CONSUMPTION limiting. While a pool exists, the SRM LIMITHARD setting cannot be changed.
3. A pool exists from the time the DEFINE RESPOOL command is issued until the z/VM system shuts down or a DELETE RESPOOL command is issued for it. The pool must be defined before SCHEDULE commands to assign guests to the pool are issued. If a permanent pool is desired, add the DEFINE RESPOOL command to the AUTOLOG1 profile or add a COMMAND statement to the AUTOLOG1 directory entry. This will ensure that the pool is created early in the IPL process, before guests permanently assigned to the pool (that is, via SCHEDULE commands in their directory entries) are logged on.
4. This command offers two very different methods for limiting CPU resources:
  - The LIMITHARD method determines the resources available to the pool based on how many shared logical cores of the specified type are online to the z/VM system. Logical processors dedicated to guest virtual CPUs are ignored. This method uses a specific percentage of the shared CPU resources for the pool, and the amount of CPU resources available to the pool will change whenever a VARY PROCESSOR command is issued to change the number of online shared cores.



**Attention:** CP always assumes that every logical core can run to 100% busy, regardless of the LPAR's entitlement and regardless of the availability of excess power on the CPC. For example, if the system has five shared logical IFL cores, CP assumes that the shared IFL CPU resource has a total capacity of 500%. If a pool is defined with a LIMITHARD 50% setting for TYPE IFL, the pool will be allowed to run to 250% busy (that is, consume up to 2.5 cores of power). If PR/SM is limiting the LPAR's five logical IFL cores to 280% total capacity, CP will still allow the pool to run to 250% busy. In that case, CP is really allowing the pool to consume  $(250/280) = 89\%$  of the scheduled IFL CPU resource.

- The CAPACITY method sets a specific CPU processor power limit for the pool. As long as the CPU CAPACITY limit is less than the amount of CPU resource of that type on the z/VM system, the amount of CPU available to the pool does not change when the VARY PROCESSOR command is issued, or if the CPU affinity of an IFL-limiting pool toggles between the on and suppressed settings. For example, a pool with a CAPACITY of 0.8 IFL cores will retain that limit regardless of the number of IFL cores on the system. If there are no IFL cores on the system, this pool will be running with CPU affinity suppressed and will be limited to 0.8 CP cores.

The only time a system change can affect the resources given to a pool with a CAPACITY limit is if the CAPACITY value is greater than the resources of that type on the system before or after the change. For example, say that the limit for LINGRP1 is 3.5 IFLs but there are only 2 IFL cores varied online. In this case, the CAPACITY limit is greater than the amount of shared CPU resource on the z/VM system, so the pool runs as if there is no CPU limit, and the LINGRP1 users can use all of the IFL cores available (2.0) without exceeding their limit. If another IFL core is varied online, this pool can now use more CPU resource (up to the available 3.0 IFLs). If a fourth IFL core is varied online, this pool will be limited to 3.5 IFL cores, and that limit will remain if more IFL cores are added.

5. You are allowed to define a resource pool with virtual CPU type IFL when there is no IFL core on that z/VM system. Users assigned to this pool are required to have CPU affinity on and will run with CPU affinity suppressed while on the z/VM system with no IFL cores. All users in the pool need to use the same underlying CPU resource for pooling to work, which is why all users assigned to a pool must have CPU affinity on. Based on the availability of IFL cores, z/VM will toggle the users and the pool together between CPU affinity on and CPU affinity suppressed.
6. If pools are used in an SSI environment, compatible pools should be defined on all members of the SSI cluster where users assigned to the resource pool could log on or be relocated. A user who is assigned to a pool can be relocated only to another member of the SSI cluster where a compatible pool exists. Compatible pools have the same pool name and the same type of virtual CPU. If a guest is being relocated between a system with IFLs and one without them, CPU affinity will be adjusted on the destination system so that the CPU affinity settings for the guest and the pool are identical.
7. After a pool has been created, use the SCHEDULE command to assign users to and remove users from the pool. When a guest is in a pool, its primary CPU type cannot be changed with the DEFINE CPU command, and its CPU affinity cannot be turned off with the SET CPUAFFINITY command.
8. When APAR VM65680 is applied and multithreading is enabled, prorated core time is used in the pool limiting calculation for all pools. When multithreading is not enabled or APAR VM65680 is not applied, raw CPU time is used instead. For an explanation of the different measures of CPU time, see [Simultaneous Multithreading \(SMT\) in z/VM: Performance](#).
9. LIMITHARD 100% is equivalent to and reported as CPU NOLIMIT.
10. Share settings control access to CPU power on a system-wide, per-processor-type basis, but they have no bearing or influence on how the power associated with a RESPOOL is distributed to its members. When the RESPOOL limit is reached, all members of the pool are put onto the limit list for a while, and then, after a delay sufficient to compensate for any excess use, they are all removed. The scheduler makes no effort to distribute the power of the RESPOOL to its members in accordance with the members' share settings.

## Responses

Response 1:

```
Resource pool poolname is created
```

confirms that the resource pool has been created.

## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

Messages:

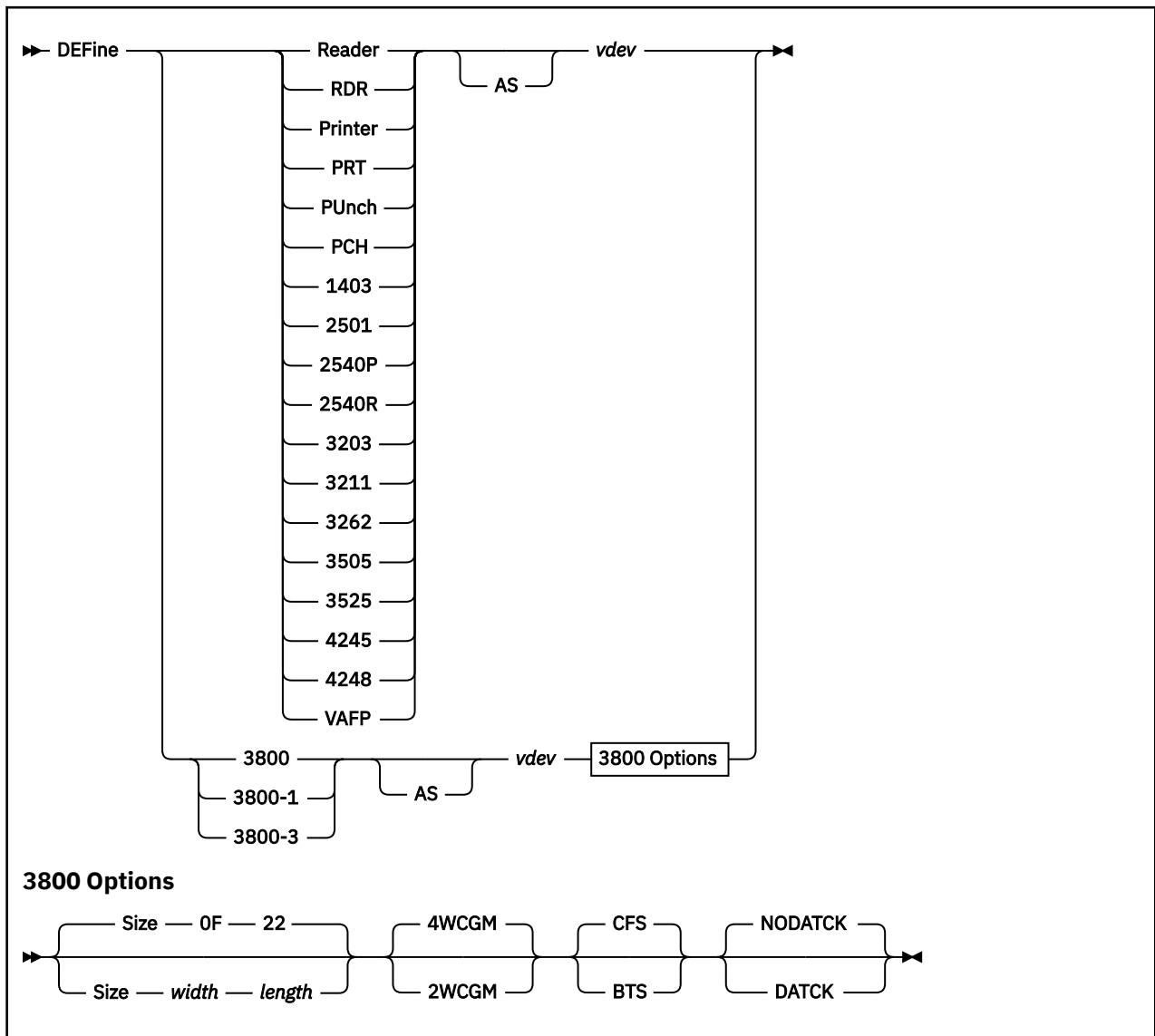
- HCP1001E An operand is missing for CPU
- HCP1001E An operand is missing for LIMITHARD
- HCP1001E An operand is missing for CAPACITY
- HCP1001E An operand is missing for RESPOOL
- HCP1001E An operand is missing for TYPE
- HCP1001E An operand is missing for STORAGE

## DEFINE RESPool

- HCP3154E Resource pool *poolname* cannot be defined unless SET SRM LIMITHARD CONSUMPTION is in effect
- HCP3155E Resource pool *poolname* is already defined



## DEFINE (Spooling Device)



### Authorization

Privilege Class: G

### Purpose

Use DEFINE (spooling device) to add pool devices to your virtual machine configuration.

### Operands

#### Reader

##### RDR

specifies a card reader.

#### Printer

##### PRT

specifies a 1403 printer.

## DEFINE (Spooling Device)

### **PUnch**

#### **PCH**

specifies a card punch.

#### **1403**

specifies a 1403 printer.

#### **2501**

specifies a 2501 card reader.

#### **2540P**

specifies a 2540 card punch.

#### **2540R**

specifies a 2540 card reader.

#### **3203**

specifies a 3203 printer.

#### **3211**

specifies a 3211 printer.

#### **3262**

specifies a 3262 printer.

#### **3505**

specifies a 3505 card reader.

#### **3525**

specifies a 3525 card punch.

#### **4245**

specifies a 4245 printer.

#### **4248**

specifies a 4248 printer.

#### **VAFP**

specifies a VAFP printer.

#### **3800**

specifies a 3800 Model 1 printer.

#### **3800-1**

specifies a 3800 Model 1 printer.

#### **3800-3**

specifies a 3800 Model 3 printer in Model 1 compatibility mode.

#### **vdev**

#### **AS vdev**

is the virtual device number for the spooling device you are defining.

#### **SIZE *width length***

specifies the physical characteristics of the paper to be loaded into the 3800 printer. The variable *width* is the hexadecimal width code of the paper. The variable *length* is the decimal length of the paper.

Specify *length* as a whole number using half-inches. If *width* and *length* are not specified, 14-7/8 x 11 inches is assumed. The default value of *width* is X'0F'. The default value of *length* is 22 (11 inches).

The following is a list of form width codes available; all other values are rejected.

<b>Code</b>	<b>Width in Inches</b>	<b>Width in Millimeters (ISO)</b>
01	6 - 1/2	165
02	Reserved	180
04	8 - 1/2	215
06	9 - 1/2	235

Code	Width in Inches	Width in Millimeters (ISO)
07	9 - 7/8	250
08	10 - 5/8	270
09	11	280
0A	12	305
0B	Reserved	322
0D	13 - 5/8	340
0E	14 - 3/10	363
0F	14 - 7/8	378

**4WCGM****2WCGM**

specifies the number of writable character generation modules (WCGMs) assumed for the virtual 3800 printer. A WCGM is a 64-position portion of 3800 character generation storage that holds the scan elements of 1 character set. A 3800 can have either 2 or 4 WCGMs. If neither is specified, the default is 4WCGM.

**Note:** Specifying 3800-3 with 2WCGM results in an error message.

**CFS****BTS**

specifies the stacker for the virtual 3800 printer. You can choose either CFS (continuous forms stacker) or BTS (burster trimmer stacker). If neither is specified, the default is CFS.

**NODATCK****DATCK**

specifies the desired handling of certain virtual 3800 data checks. If neither is specified, the default is NODATCK.

If you specify DATCK, all 3800 data checks are reflected to the virtual machine (provided the BLOCK DATA CHECK CCW has not been issued). If you specify NODATCK, only data checks that occur because of invalid translate table specifications or unmatched FCB codes are reflected to the virtual machine.

**Note:** DATCK should be specified only when absolutely necessary, as it severely increases the overhead associated with simulation of WRITE and SKIP CCWs to the virtual 3800. In general, the reflection of data checks because of overprinting and invalid EBCDIC codes is not necessary.

**Usage Notes**

1. Avoid using device addresses that can be mistaken for device types (such as 1403 and 3203). When it encounters a device address that is the same as an IBM device type, CP assumes that you are specifying a device type and not a device address.
2. If you define a virtual 3211 printer, it supports the CP command LOADVFCB and the INDEX feature.
3. The default classes given to spool files created on this device are as follows:
  - Class A reader
  - Class A printer
  - Class A punch.
4. Virtual printers, consoles, and punches are created with the default option NOEOF.
5. Defaults for some real printer attributes (for example CHARS) are not set when a virtual printer is created with the DEFINE command.
6. The maximum data length for a VAFP spool file is 32767 bytes. See the RSCS library for information on RSCS logical record length restrictions.

## DEFINE (Spooling Device)

7. There is no Channel Command Word (CCW) validation performed for VAFP devices. Therefore, spool files for VAFP devices may contain CCWs that do not appear for other devices (for example: SENSE ID and READ FCB). Specifically, when performing CCW I/O operations to a VAFP printer, most CCWs in the channel program are treated as a write operation CCW. The exceptions are X'08' which is treated as a transfer in channel command, X'03' with a byte count of one which is treated as a no operation instruction, and X'04' which is treated as a sense command.

The flags in the CCW flag field (for example, command chaining and data chaining flags) will be respected for VAFP devices as they are for other virtual printers with one exception. The SKIP flag will be ignored because most CCWs are treated as write operations by these devices. The SKIP flag will be preserved in the CCW that is placed in the spool file.

8. VAFP spool files with skip to channel other than 1 will not be rejected.
9. There is no corresponding real device type for a VAFP.
10. VAFP spool files are intended for use by AFP printers and real printers managed by RSCS. VAFP printers are not intended for normal use as a virtual printer.

Print files created on VM using the VAFP support can be chosen for printing by the CP real printer system. Care must be taken by the application which creates these files. If the file is to be printed on a VM system printer, the application must only use standard impact printer CCWs to create the file.

11. Files created on a VAFP device can be read by the virtual card reader using either an assembler I/O instruction or DIAGNOSE code X'A8'. However, the data in these files may be truncated. With respect to the read operation, these files are treated like a file created on a virtual 3800 printer. That is, the file is treated as if it has a logical record length of 204 and the records are truncated at that point. The data in these files can be accessed without truncation by using DIAGNOSE X'14' or the IUCV \*SPL system service.
12. For more information, see [“Usage Notes” on page 122](#).

### Examples

```
define reader as 00c
```

You can specify the operands after *vdev* in any order and repeatedly. However, the last specification of a repeatedly defined operand is the one used. For example, if you enter:

```
define 3800 as 00a cfs datck size 07 22 2wcgm bts
```

BTS is the value used instead of CFS because it was specified after CFS.

### Responses

Response 1:

```
type vdev DEFINED
```

confirms that the device has been defined.

#### **type**

identifies a reader, printer, or punch.

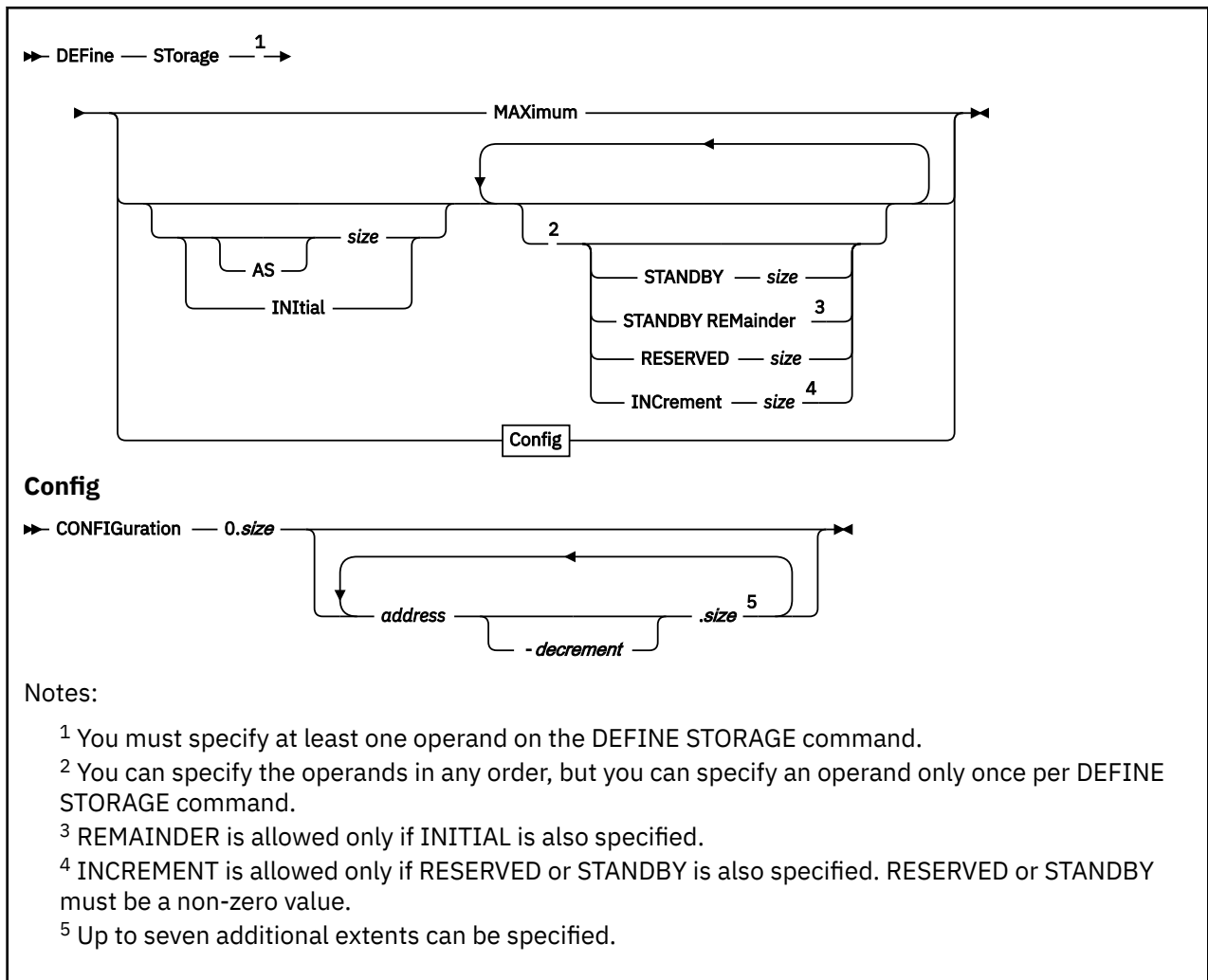
#### **vdev**

identifies the virtual number of the spool device.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE STORAGE



### Authorization

Privilege Class: G

### Purpose

Use DEFINE STORAGE to change the storage size of your virtual machine.

### Operands

#### MAXIMUM

specifies that the maximum size of virtual machine storage as defined on the MAXSTORAGE statement, if any, or defined in the USER directory entry for the virtual machine, is to be the online storage size of your virtual machine. No other keywords can be specified, because when MAXIMUM is specified, the RESERVED and STANDBY storage sizes are 0.

#### AS size

specifies the online storage size of your virtual machine. When the RESERVED or STANDBY operand is also specified, the AS size might be adjusted upward because of the storage-increment size specified by the INCREMENT operand, or as determined by CP. The sum of the AS or INITIAL, RESERVED, and STANDBY storage sizes cannot exceed the maximum storage size specified for the virtual machine in the CP directory.

## DEFINE STORAGE

See usage note [“8” on page 217](#) for an example that demonstrates how the AS size might be adjusted by CP.

### **INITIAL**

specifies that the initial size of virtual machine storage as defined on the STORAGE statement, if any, or defined in the USER directory entry for the virtual machine, is to be the online storage size of your virtual machine. When the RESERVED or STANDBY operand is also specified, the INITIAL size might be adjusted upward because of the storage-increment size specified by the INCREMENT operand, or as determined by CP. The sum of the AS or INITIAL, RESERVED, and STANDBY storage sizes cannot exceed the maximum storage size specified for the virtual machine in the CP directory.

IBM recommends that you use INITIAL in a DEFINE STORAGE command with the STANDBY REMAINDER option specified.

See usage note [“14” on page 218](#) to understand in which situations using INITIAL causes a system-reset-clear function to be performed.

### **STANDBY size**

specifies the standby storage size of your virtual machine.

The STANDBY size might be adjusted upward by CP because of the storage-increment size specified by the INCREMENT operand, or as determined by CP. The sum of the AS or INITIAL, RESERVED, and STANDBY storage sizes cannot exceed the maximum storage size specified for the virtual machine in the CP directory. You can specify a STANDBY storage size of 0.

See usage note [“1” on page 216](#) for more information about using STANDBY storage, and see usage note [“8” on page 217](#) for an example of how the STANDBY storage size might be adjusted by CP.

### **STANDBY REMAINDER**

specifies that the STANDBY storage size should be calculated. The INITIAL keyword must be specified when you specify STANDBY REMAINDER. When you specify STANDBY REMAINDER, the STANDBY storage is equal to the maximum storage size (from the CP directory) minus the initial storage size (from the CP directory) and the RESERVED value specified on the DEFINE STORAGE command. The STANDBY size might be adjusted upward or downward by CP because of the storage-increment size specified by the INCREMENT operand, or as determined by CP. The sum of the AS or INITIAL, RESERVED, and STANDBY storage sizes cannot exceed the maximum storage size specified for the virtual machine in the CP directory.

See usage note [“13” on page 218](#) for examples of using the STANDBY REMAINDER keyword.

### **RESERVED size**

specifies the reserved storage size of your virtual machine.

The RESERVED size might be adjusted upward by CP because of the storage-increment size specified by the INCREMENT operand, or as determined by CP. The sum of the AS or INITIAL, RESERVED, and STANDBY storage sizes cannot exceed the maximum storage size specified for the virtual machine in the CP directory. You can specify a RESERVED storage size of 0.

See usage note [“14” on page 218](#) to understand in which situations using RESERVED causes a system-reset-clear function to be performed.

### **INCREMENT size**

specifies the size of a unit of storage that can be added or removed by an operating system running in a virtual machine. The minimum size is 1 MB and must be a power of 2. If necessary, CP adjusts the value up or down to conform to the architected requirements.

See usage note 8 for an example demonstrating how the INCREMENT size might be adjusted by CP.

When no standby or reserved storage is defined for the virtual machine, by default or because the STANDBY and RESERVED operands are set to 0, the INCREMENT operand is not allowed. In this case, the increment size reflected to a second-level operating system is 1 MB.

See usage note [“14” on page 218](#) to understand in which situations using INCREMENT causes a system-reset-clear function to be performed.

**size**

is the size of the storage you are defining (when CONFIGURATION is not specified). The *size* value is of the form *nu*, where *n* is a 1- to 7-digit decimal number and *u* is the 1-character storage unit suffix (see Table 17 on page 215). For the AS value, *n* must be greater than 0. For the RESERVED or STANDBY value, a size of 0 is valid.

Table 17. Maximum Input Values for Storage Units

Storage unit (powers of 2)	Suffix ( <i>u</i> )	Maximum input value ( <i>n</i> )
Kilobytes	K	2096128
Megabytes	M	9999999
Gigabytes	G	9999999
Terabytes	T	9999999
Petabytes	P	16384
Exabytes	E	16

**Notes:**

1. The K suffix is provided for upward compatibility only; a K specification is rounded up to a MB value. The maximum specification is 2096128K (2047 MB). The K suffix is not valid with a CONFIGURATION specification.
2. The maximum input value of 9999999 for the M, G, or T suffix is not a size limit but the physical limit of the operand (7 digits plus suffix). If the maximum input value for one of these suffixes does not allow you to define the amount of storage you want, you need to use a larger storage unit.
3. An XC virtual machine cannot address more than 2047 MB of storage in its base address space.

**CONFIGURATION**

allows you to define multiple, noncontiguous storage extents. You can also define a single extent just as you would without CONFIGURATION, for example:

```
DEF STOR 64M
DEF STOR CONFIG 0.64M
```

The advantage of CONFIGURATION is that it provides more detail in the command response and in the output of the QUERY VIRTUAL STORAGE command.

**0.size**

is the size of the first extent you are defining (when CONFIGURATION is specified), which always starts at address 0. The *size* value is of the form *nu*, where *n* is a 1- to 7-digit decimal number and *u* is the 1-character storage unit suffix (see Table 17 on page 215).

**address**

is the starting address of the extent you are defining. The *address* value is of the form *nu*, where *n* is a 1- to 7-digit decimal number and *u* is the 1-character storage unit suffix (see Table 17 on page 215). The first address must be zero. Subsequent extent start specifications must specify a higher address than the ending address of the last extent plus one (contiguous extents are not valid).

**.size**

is the size of the extent you are defining. The *size* value is of the form *nu*, where *n* is a 1- to 7-digit decimal number and *u* is the 1-character storage unit suffix (see Table 17 on page 215).

**-decrement**

is subtracted from the extent starting *address* specified to calculate the actual extent start address you are defining. The *decrement* value is of the form *nnnu*, where *nnnn* is a 1- to 4-digit decimal number and *u* is the 1-character storage unit suffix (see Table 17 on page 215). The *decrement*:

- Cannot be larger than the *address*
- Cannot cause overlap or abutment of a previous extent specification.

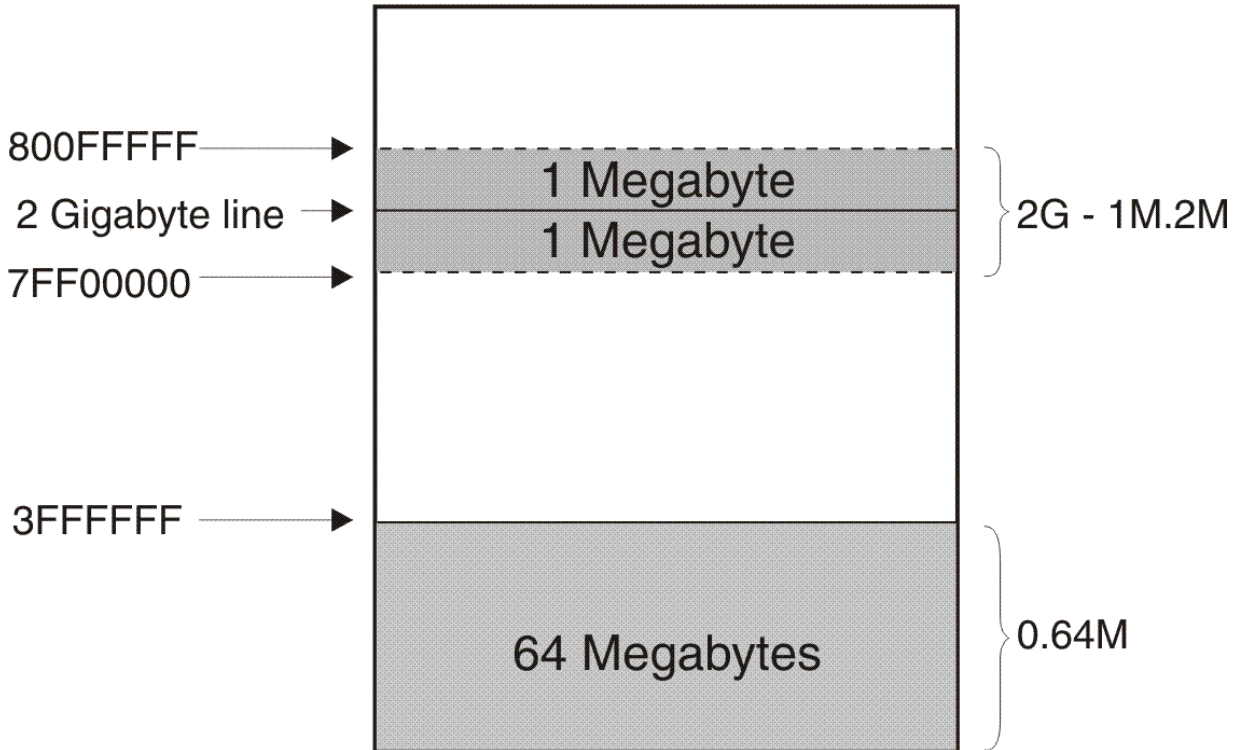
## DEFINE STORAGE

- Must use a unit of measure smaller than the one used for *address*

The *decrement* value helps to determine addresses of higher values, especially when they cross boundaries, for example:

```
DEF STOR CONFIG 0.64M 2G-1M.2M
```

This would define the 66 MB of storage shown below without you having to know (or type in) the address that is 1 MB below the 2 GB line:



The response would be:

```
def stor config 0.64m 2g-1m.2m
STORAGE = 66M
Storage Configuration:
0.64M 2G-1M.2M
Extent Specification          Address Range
-----
          0.64M              0000000000000000 - 0000000003FFFFFF
          2G-1M.2M          000000007FF00000 - 00000000800FFFFF
Storage cleared - system reset.
```

See Response 3 for a detailed definition of the information.

## Usage Notes

1. The STANDBY operands allow you to define standby storage before you IPL a guest operating system in the virtual machine.

This allows a second-level operating system to exploit the dynamic storage reconfiguration (DSR) capabilities of the architecture. For example, the operator of a second-level z/VM system can issue the SET STORAGE command to convert standby storage to configured storage, thereby increasing the size of the storage configuration, or convert configured storage to standby storage to reduce the size of the storage configuration. These changes can be seen first-level in the QUERY VIRTUAL STORAGE command response values for STORAGE= and STANDBY=.

2. AS specified alone (without a RESERVED or STANDBY value) defines the online storage size of the virtual machine. If the online storage size is changed, or if reserved or standby storage was previously



defined, a system-reset-clear function is performed. If reserved or standby storage was previously defined, their sizes are set to zero.

3. RESERVED specified alone (without an AS or STANDBY value) defines the reserved storage size of the virtual machine but does not change the online size or standby size. You can specify a reserved storage size of 0, with or without an AS or STANDBY value. A system-reset-clear function is performed when a RESERVED value is specified. Use of RESERVED is not recommended unless directed by IBM.
4. STANDBY specified alone (without an AS or RESERVED value) takes the standby amount of storage from the reserved storage size or returns it. That is, the RESERVED value is implicitly adjusted opposite to the STANDBY value, so that their sum remains unchanged. When you specify only the STANDBY value, it cannot be set to greater than the sum of the current STANDBY and RESERVED values. No system-reset-clear function is performed when a STANDBY value is specified alone.
5. If more than one option is specified, a new virtual machine configuration is defined containing the respective storage areas:
  - If the AS option is not specified, its value is the previously configured amount. If a virtual machine has changed the amount of its real storage (for example, if the operator of a second-level z/VM system has issued the SET STORAGE command or a Linux guest has issued a chmem command), the response for a DEFINE STORAGE or QUERY VIRTUAL STORAGE command at first level will reflect the updated amount of online configured and STANDBY storage for the virtual machine.
  - If AS is specified and either the RESERVED or STANDBY option is not specified, the respective value is assumed to be zero.

If any storage size is different from the current configuration, a system-reset-clear function is performed.

6. If the DEFINE STORAGE command performs a system-reset-clear function, it is as though the SYSTEM CLEAR command had been processed. The effects of this system-reset-clear function are described in the description of the CLEAR operand of the SYSTEM command. (See “SYSTEM” on page 1911.) Because a system-reset-clear function sets all of your virtual machine storage to binary zeros, you must repeat your IPL procedure to continue operation.
7. A DEFINE STORAGE command that results in a system-reset-clear function will fail when issued while a VMRELOCATE of the virtual machine is in progress.
8. A storage increment is the unit in which an operating system can add or remove storage. If the INCREMENT operand is not specified on the DEFINE STORAGE command, CP determines the increment size. When no standby or reserved storage is defined for the virtual machine, the increment size is set to 1 MB. Otherwise, the increment size must be a power of 2 in megabytes and no less than 1/512th of the AS size, or 1/512th of the sum of the STANDBY plus RESERVED sizes, whichever is larger. If the increment size you specify does not adhere to these rules, CP adjusts the value, along with the AS or INITIAL, RESERVED, and STANDBY sizes, to be multiples of the increment size. The resulting values are displayed in the DEFINE STORAGE command response. For example, in the following command, all values are adjusted:

```
DEFINE STORAGE 1M STANDBY 3G RESERVED 543G INCREMENT 3M
STORAGE = 2G MAX = 15E INC = 2G STANDBY = 4G RESERVED = 544G
```

where:

- The INCREMENT size is adjusted downward to a power of 2, namely 2 MB.
- The INCREMENT size is adjusted again to 1/512th of the AS size, and made a power of 2; in this case it remains 2 MB.
- The INCREMENT size is adjusted again to 1/512th of STANDBY + RESERVED, and made a power of 2, specifically 2 GB.
- The AS size must be adjusted to at least the INCREMENT size, and must be a multiple of the INCREMENT size, specifically 2 GB.
- The STANDBY size must be adjusted to at least the INCREMENT size, and must be a multiple of the INCREMENT size, specifically 4 GB

## DEFINE STORAGE

- The RESERVED size must be adjusted to at least the INCREMENT size, and must be a multiple of the INCREMENT size, specifically 544 GB.

If any adjustment causes the sum of the AS or INITIAL, RESERVED, and STANDBY sizes to exceed the maximum storage size specified for the virtual machine in the CP directory, the command fails and message HCP0094E is issued.

9. If the increment size you specify on the DEFINE STORAGE command using the INCREMENT operand, or the increment size determined by the DEFINE STORAGE command, is less than 128 MB, a second-level z/VM system running in the virtual machine will apply a dynamic storage reconfiguration (DSR) unit size of 128 MB in place of the storage-increment size. When this occurs, the increment size (INC=) displayed using the QUERY STORAGE command on the second-level z/VM system is different from the increment size (INC=) displayed by the QUERY VIRTUAL command first-level.

Because storage must be added or removed from an operating system's storage configuration in units of the storage-increment size, z/VM's usage of the DSR unit size as the effective storage-increment size, can make a portion of a virtual storage configuration unusable by a second-level z/VM system. In addition, storage values that normally round up to a storage-increment multiple, will be rounded down when insufficient CONFIGURED or STANDBY storage exists to round up.

10. The CONFIGURATION option with more than one extent is primarily for testing. Do not confuse this with loading discontinuous saved segments. Directory limits apply to the sum of the extent sizes.
11. When a second-level z/VM system comes up with the STORE=*n* IPL parameter in a virtual machine whose storage was defined with a DEFINE STORAGE CONFIGURATION command, and the STORE=*n* value falls outside of a storage extent, the highest storage address online after IPL is equal to the value that would have resulted if the STORE=*n* value had been specified.
12. For general usage notes for the DEFINE command, see [“Usage Notes” on page 122](#).
13. An example of using the STANDBY REMAINDER keyword follows:

- With this directory entry:

```
USER ANYONE SECRET18 6G 48G ABCDEFG
```

with this command:

```
DEFINE STORAGE INITIAL STANDBY REMAINDER RESERVED 16G
```

the response would be:

```
STORAGE = 6G MAX = 48G INC = 128M STANDBY = 26G RESERVED = 16G
```

14. A system-reset-clear function is performed when any of these conditions are met:

- The INCREMENT, INITIAL, or RESERVED operand is specified.
- The requested AS size is different than the current configured value.
- The requested configuration has neither STANDBY nor RESERVED storage, but the current configuration has STANDBY storage, RESERVED storage, or both. For example:

Given the current memory configuration of:

```
STORAGE = 16G MAX = 32G INC = 32M STANDBY = 5G RESERVED = 5G
```

with this command:

```
DEFINE STORAGE 16G
```

the response would be:

```
STORAGE = 16G  
Storage cleared - system reset
```

- The requested configuration has STANDBY storage, but the current configuration has no STANDBY storage. For example:

Given the current memory configuration of:

```
STORAGE = 16G
```

with this command:

```
DEFINE STORAGE 16G STANDBY 5G
```

the response would be:

```
STORAGE = 16G MAX = 32G INC = 32M STANDBY = 5G RESERVED = 0
Storage cleared - system reset
```

- The requested configuration's sum of STANDBY and RESERVED storage is different from the current configuration's sum of STANDBY and RESERVED storage. For example:

Given the current memory configuration of:

```
STORAGE = 16G MAX = 32G INC = 32M STANDBY = 5G RESERVED = 5G
```

with this command:

```
DEFINE STORAGE 16G STANDBY 7G
```

the response would be:

```
STORAGE = 16G MAX = 32G INC = 32M STANDBY = 7G RESERVED = 0
Storage cleared - system reset
```

## Responses

Response 1 is displayed when the DEFINE STORAGE command contains only the AS, INITIAL, or MAXIMUM operand:

```
STORAGE = nu
```

where:

### **STORAGE = nu**

specifies the size of virtual machine storage.

Response 2 is displayed when the DEFINE STORAGE command contains the STANDBY operand, the RESERVED operand, or both:

```
STORAGE = nu MAX = nu INC = nu STANDBY = nu RESERVED = nu
```

where:

### **STORAGE = nu**

specifies the size of online (assigned) virtual machine storage.

### **MAX = nu**

specifies the maximum size of virtual machine storage as defined in the user directory entry for the virtual machine. You cannot use the DEFINE STORAGE command to increase the virtual machine's total storage beyond this amount.

### **INC = nu**

specifies the size of the virtual machine storage increment.

### **STANDBY = nu**

specifies the amount of virtual machine storage in standby state that is available to be configured online.

### **RESERVED = nu**

specifies the amount of virtual machine storage in reserved state that can be defined online or as standby. Use of RESERVED is not recommended unless directed by IBM.

## DEFINE STORAGE

Response 3 is displayed when the DEFINE STORAGE command results in a system-reset-clear function:

```
STORAGE CLEARED - SYSTEM RESET
```

Response 4 is displayed when the DEFINE STORAGE command contains the CONFIGURATION operand:

```
STORAGE = nu
Storage Configuration:
configspec

Extent Specification                Address Range
-----
          0.ssssssssssssu 00000000_00000000 - endadres_endadres
boundaryaddressu.ssssssssssssu begadres_begadres - endadres_endadres
boundaryadru-nnnnu.ssssssssssssu begadres_begadres - endadres_endadres
```

where:

### **STORAGE = nu**

specifies the size of online (assigned) virtual machine storage, including the configuration extents.

### **configspec**

is the storage configuration specification input used on the DEFINE STORAGE CONFIGURATION command that defines the current storage configuration.

### **Extent Specification**

specifies the information supplied by the QUERY VIRTUAL STORAGE command. Each configuration extent is listed on a separate line:

#### **boundaryaddressu**

#### **boundaryadru**

is the boundary address as specified in the DEFINE STORAGE CONFIGURATION command, where *u* is the 1-character storage unit suffix.

#### **sssssssssssssu**

is the extent size, 1 - 14 decimal digits followed by the 1-character storage unit suffix.

#### **nnnnu**

is the decrement specified in DEFINE STORAGE CONFIGURATION used to modify the extent start boundary address, expressed as 1 - 4 decimal digits followed by the 1-character storage unit suffix (which must be a smaller storage unit than that used for the start address it modifies). The decrement is subtracted from the base start boundary specification to determine the effective extent start boundary address.

### **Address Range**

specifies the extent information converted to beginning and ending addresses:

#### **begadres\_begadres**

is the 16-digit extent start address expressed in hexadecimal format (always ends in X'00000', a megabyte boundary). Underscores are used for readability if SET UNDERSCORE ON has been issued.

#### **endadres\_endadres**

is the 16-digit extent end address expressed in hexadecimal format (always ends in X'FFFFFF', 1 less than a megabyte boundary). Underscores are used for readability if SET UNDERSCORE ON has been issued.

## Messages

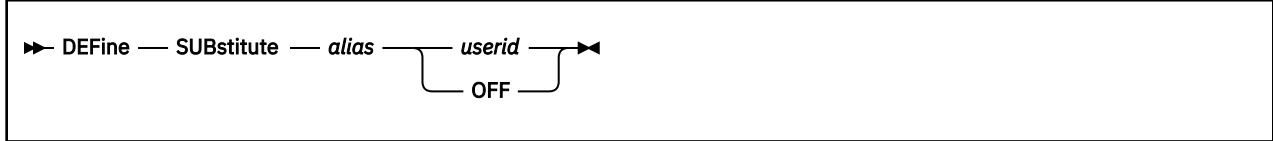
For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

Messages:

- HCP093E RESERVED (*resize*) exceeds hardware maximum (*maxsize*)
- HCP093E STANDBY (*stbysize*) exceeds hardware maximum (*maxsize*)
- HCP093E STANDBY (*stbysize*) plus RESERVED (*resize*) exceeds hardware maximum (*maxsize*)

- HCP093E Storage size (*storsize*) exceeds hardware maximum (*maxsize*)
- HCP093E Storage size (*storsize*) plus RESERVED (*ressize*) exceeds hardware maximum (*maxsize*)
- HCP093E Storage size (*storsize*) plus STANDBY (*stbysize*) exceeds hardware maximum (*maxsize*)
- HCP093E Storage size (*storsize*) plus STANDBY (*stbysize*) plus RESERVED (*ressize*) exceeds hardware maximum (*maxsize*)
- HCP094E Storage size (*storsize*) plus STANDBY (*stbysize*) plus RESERVED (*ressize*) exceeds directory maximum (*maxsize*)
- HCP096E Configuration extent (*extent*) exceeds maximum allowed on this processor (*maxsize*). Storage configuration remains unchanged.
- HCP098E Insufficient storage available to satisfy STANDBY request
- HCP099E {STANDBY|RESERVED|INCREMENT} option conflicts with current storage configuration

## DEFINE SUBSTITUTE



### Authorization

Privilege Class: A

### Purpose

Use DEFINE SUBSTITUTE to define which virtual machine is actually used as the target for IUCV communication requests addressed to a specified *alias* user identifier.

### Operands

#### *alias*

is the substitute to be defined. The *alias* may not be the user identifier of a virtual machine that is defined in the CP User Directory.

#### *userid*

specifies the user identifier for which the substitute is to be set or cleared.

#### OFF

removes the substitute for the user.

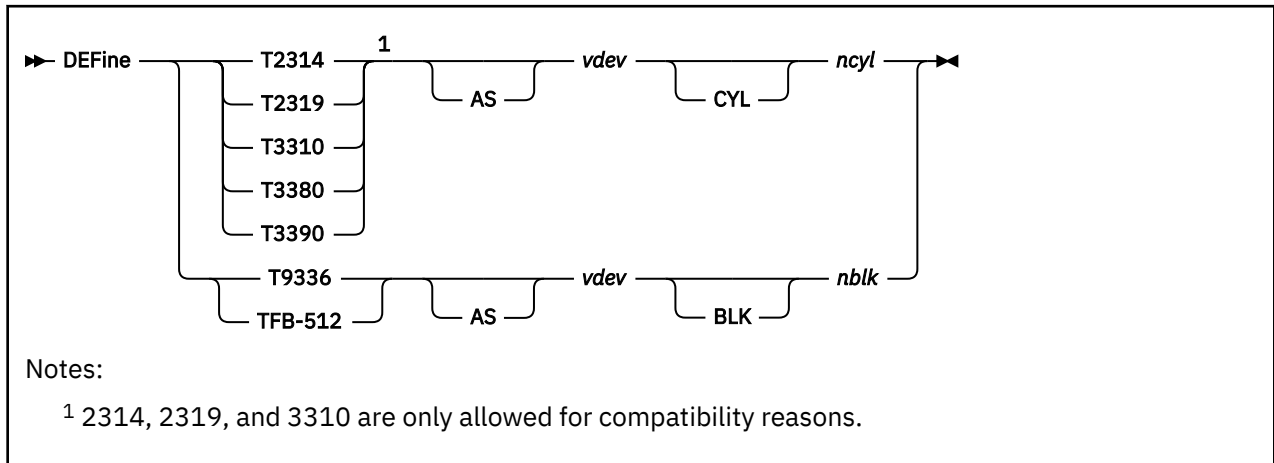
### Usage Notes

1. A substitute may be used to refer to a virtual machine by another name (user identifier) for the purposes of inter-machine communication using IUCV. Any communications addressed to the substitute will be redirected to the specified virtual machine.
2. A substitute may be defined or removed for a virtual machine regardless of whether it is logged on. If a substitute is removed, existing IUCV communication paths that were established using it are not affected.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE (Temporary Disk)



### Authorization

Privilege Class: G

### Purpose

Use DEFINE (temporary disk) to increase virtual workspace by attaching temporary disks to your virtual machine. A temporary disk is private (cannot be shared by another virtual machine).

### Operands

#### T2314, T2319, or T3310

All above DASD are accepted for compatibility but result in a message indicating that the temporary disk is not defined because space is not available.

#### T3380, T3390, T9336, or TFB-512

All above DASD add a temporary virtual disk to your virtual machine configuration. The 4 digits following the T correspond to the virtual device type. TFB-512 indicates that any supported FBA device is acceptable.

#### AS *vdev*

##### *vdev*

is the virtual device number of the temporary disk you are defining.

#### CYL *ncyl*

##### *ncyl*

is the size of this temporary disk in cylinders (valid only for CKD and ECKD disks).

#### BLK *nblk*

##### *nblk*

is the size of this temporary disk in blocks (valid only for FBA disks).

### Usage Notes

1. Temporary disk space is assigned from a pool of DASD resources; therefore, you should always format your temporary disk space before you use it.
2. The temporary disk may be slightly larger than the size requested because of CP's implementation of system temporary disk space management.
3. Using ICKDSF is the preferred method for formatting a DASD device if it is to be used by CP (such as an IPL device). The CMS FORMAT command should be used to format a DASD device if it is to be used as a CMS disk.

## DEFINE (Temporary Disk)

4. Temporary disks created with this command have cache access if they are on a cached control unit.
5. A minidisk, including a temporary disk, is not eligible for minidisk cache if it has been defined with greater than 32767 cylinders. This does not apply to FBA devices. It applies only to ECKD devices.
6. If you define a temporary disk as virtual device number 192, the following special rules apply when you IPL CMS:
  - If 192 is unformatted, CMS formats it and accesses it as file mode D.
  - If 192 is CP-formatted, CMS reformats it for CMS use and accesses it as file mode D.
  - If 192 is CMS-formatted and accessed as a file mode other than D, CMS reaccesses it as file mode D.When CMS accesses a 192 minidisk as file mode D, any minidisk or SFS directory already accessed as D is released.
7. For more information, see [“Usage Notes” on page 122](#).

### Responses

```
DASD vdev DEFINED
```

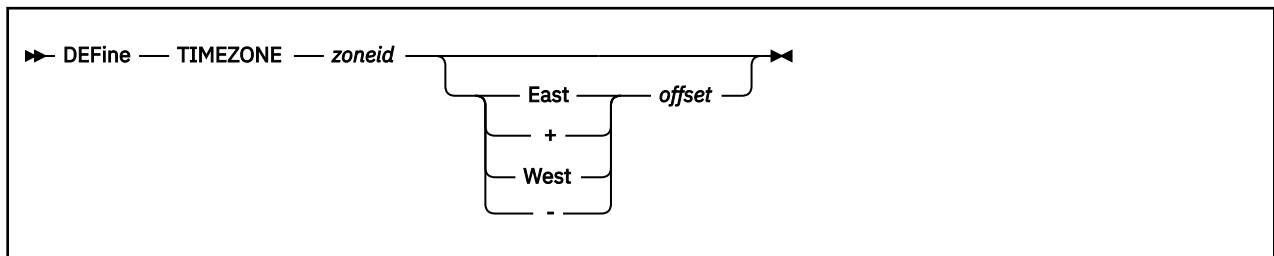
confirms the definition of the temporary disk.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).



## DEFINE TIMEZONE



### Authorization

Privilege Class: A

### Purpose

Use DEFINE TIMEZONE to define a new time zone or change an existing time zone definition.

To activate a time zone, use the SET TIMEZONE command. To display the defined time zones, use the QUERY TIMEZONES command. For more information, see [“SET TIMEZONE” on page 1738](#) and [“QUERY TIMEZONES” on page 1222](#).

### Operands

#### *zoneid*

is the time zone ID you wish to define or change. The time zone ID must be an alphanumeric string of 1-4 characters. GMT (Greenwich Mean Time) and UTC (Coordinated Universal Time) are reserved time zone IDs.

#### **East**

+

indicates that the time zone is east of longitude 0. CP adds the specified *offset* to UTC to define the time zone ID. EAST is the same as +.

#### **West**

-

indicates that the time zone is west of longitude 0. CP subtracts the specified *offset* from UTC to define the time zone ID. WEST is the same as -.

#### *offset*

is the time difference between UTC and the zone ID you are defining. You can specify the offset in either of the following formats:

*hh:mm:ss*

*hh.mm.ss*

where *hh* is the hour, *mm* are the minutes, and *ss* are the seconds. You can separate the hour from the minutes and seconds using either a colon (:) or a period (.), but you cannot use both in one offset definition. You must specify at least a 1-digit number for the hour; everything else is optional. However, if you want to specify seconds, you must specify the hour and the minutes. If you want to specify the minutes, you must specify the hour.

For example, if you want to specify an offset of 5 hours, no minutes, and no seconds, all of the following are valid offsets:

```

5
05
5:0
05.00

```

## DEFINE TIMEZONE

```
5.0.0  
05:00:00
```

### Usage Notes

1. If you specify an existing time zone ID, CP changes the specified time zone definition to reflect the new direction and offset from UTC.
2. If you change the time zone definition for the zone ID that is currently in effect, you do not affect the local time on your system. To put the change into effect, use the SET TIMEZONE command. For more information on the SET TIMEZONE command, see [“SET TIMEZONE” on page 1738](#).
3. You can also define and set time zones in the system configuration file by using the TIMEZONE\_DEFINITION and TIMEZONE\_BOUNDARY statements. For more information, see [TIMEZONE\\_DEFINITION Statement](#) and [TIMEZONE\\_BOUNDARY Statement](#).

### Responses

Response 1:

When you enter the DEFINE TIMEZONE command, CP responds with information on the newly defined or changed time zone ID. For example:

```
define timezone est west 05.00.00  
Time zone EST has been defined as 05.00.00 west of UTC.  
Ready;
```

shows that you have defined or changed the time zone ID EST to be 5 hours west of UTC.

### Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

Messages:

- HCP0987E DEFINE TIMEZONE not valid - STP timezone in use

## DEFINE (Virtual Device)



### Authorization

Privilege Class: G

### Purpose

Use DEFINE (virtual device) to change the device number of any device in your virtual machine configuration.

### Operands

#### **vdev1**

is the device number of an existing virtual device.

#### **AS vdev2**

#### **vdev2**

is the new virtual device number.

### Usage Notes

1. Avoid using device addresses that can be mistaken for device types (such as 1403 and 3203). When it encounters a device address that is the same as an IBM device type, CP assumes that you are specifying a device type and not a device address. To avoid potential confusion and errors, use the REDEFINE command.
2. Virtual paging subsystem devices cannot be redefined. This is done to protect the integrity of the user virtual paging subsystem.
3. Devices that are part of a virtual network adapter cannot be redefined. Virtual network adapters are defined with the DEFINE NIC command.
4. If the existing virtual device is part of a Concurrent Copy (CONCOPY) session for your virtual machine, the CONCOPY session is terminated in the storage control associated with the device.
5. If you change the virtual device number of your console while spooling your console, the spool file device number will also change.
6. If you define a minidisk as virtual device number 192, the following special rules apply when you IPL CMS:
  - If 192 is an unformatted temporary minidisk or virtual disk in storage, CMS formats it and accesses it as file mode D.
  - If 192 is a CP-formatted temporary minidisk or virtual disk in storage, CMS reformats it for CMS use and accesses it as file mode D.
  - If 192 is a CMS-formatted temporary minidisk, virtual disk in storage, or permanent minidisk accessed as a file mode other than D, it is released.
  - If 192 is an unformatted or CP-formatted permanent minidisk, CMS does not automatically format, reformat, access, or reaccess it.

When CMS accesses a 192 minidisk as file mode D, any minidisk or SFS directory already accessed as D is released.

7. For more information, see [“Usage Notes” on page 122](#).

### Responses

Response 1:

```
type vdev2 DEFINED
```

confirms that the device number has been changed.

#### **type**

is device type you defined. The *type* values are:

#### **Type**

#### **Meaning**

#### **CONS**

Console

#### **CTCA**

Channel-to-channel adapter

#### **CTLR**

3745 Communications Controller

#### **DASD**

Direct access storage device

#### **DEV**

Any other device

#### **GRAF**

Display graphic device

#### **LINE**

Communication line

#### **MSC**

Mass Storage Controller

#### **OSA**

Open Systems Adapter

#### **PRT**

Printer

#### **PUN**

Card punch

#### **RDR**

Card reader

#### **SWCH**

Dynamic Switching Device

#### **TAPE**

Magnetic tape

#### **vdev2**

identifies the new device number.

Response 2:

```
Redefining vdev1 has caused the termination of  
CONCOPY session(s) on device(s): vdev...
```

indicates that Concurrent Copy (CONCOPY) sessions have been terminated during the process of changing the existing virtual device number.

#### **vdev1**

is the existing virtual device number that you are changing.

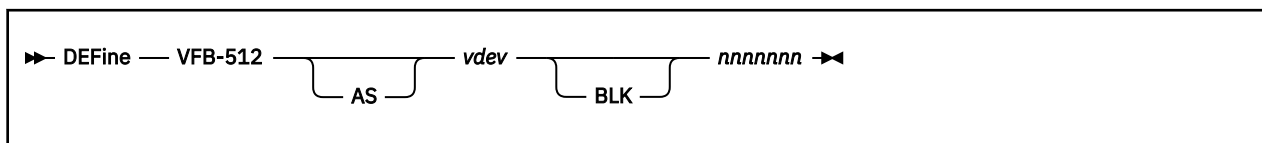
#### **vdev...**

is a list of the virtual device numbers that are part of the terminated CONCOPY sessions.

## Messages

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

## DEFINE (Virtual Disk in Storage)



### Authorization

Privilege Class: E, G

### Purpose

Use DEFINE to create a virtual disk in storage attached to your virtual machine. A virtual disk in storage created using this command is temporary, private (cannot be shared by another virtual machine), and simulates an FBA minidisk, which is defined in 512-byte blocks. The virtual disk in storage is destroyed when you detach it or log off.

### Operands

*vdev*

**AS** *vdev*

is the virtual device number to be assigned to the virtual disk in storage.

*nnnnnnn*

**BLK** *nnnnnnn*

is the size of the virtual disk in storage in 512-byte blocks. The maximum size is 4194296 blocks.

### Usage Notes

1. A virtual disk in storage is allocated from system storage, not mapped to a real DASD. Therefore, defining a virtual disk in storage does not require having a real FBA DASD in your system configuration.
2. A virtual disk in storage must be formatted before it can be used. Using ICKDSF is the preferred method for formatting the virtual disk in storage if it is to be used as a system disk by a guest (such as VSE) or a second-level CP. The CMS command FORMAT should be used to format the virtual disk in storage if it is to be used as a CMS disk.
3. Virtual disks in storage are allocated in 8-block pages. Therefore, the size of the virtual disk in storage that is created may be rounded up to the nearest page. You can use all of the blocks in the created virtual disk in storage.
4. If you define a virtual disk in storage as virtual device number 192, the following special rules apply when you IPL CMS:
  - If 192 is unformatted, CMS formats it and accesses it as file mode D.
  - If 192 is CP-formatted, CMS reformats it for CMS use and accesses it as file mode D.
  - If 192 is CMS-formatted and accessed as a file mode other than D, CMS reaccesses it as file mode D.

When CMS accesses a 192 minidisk as file mode D, any minidisk or SFS directory already accessed as D is released.
5. This command will fail if issued when VMRELOCATE is in progress.
6. For more information, see [“Usage Notes” on page 122](#).

### Responses

Response 1:

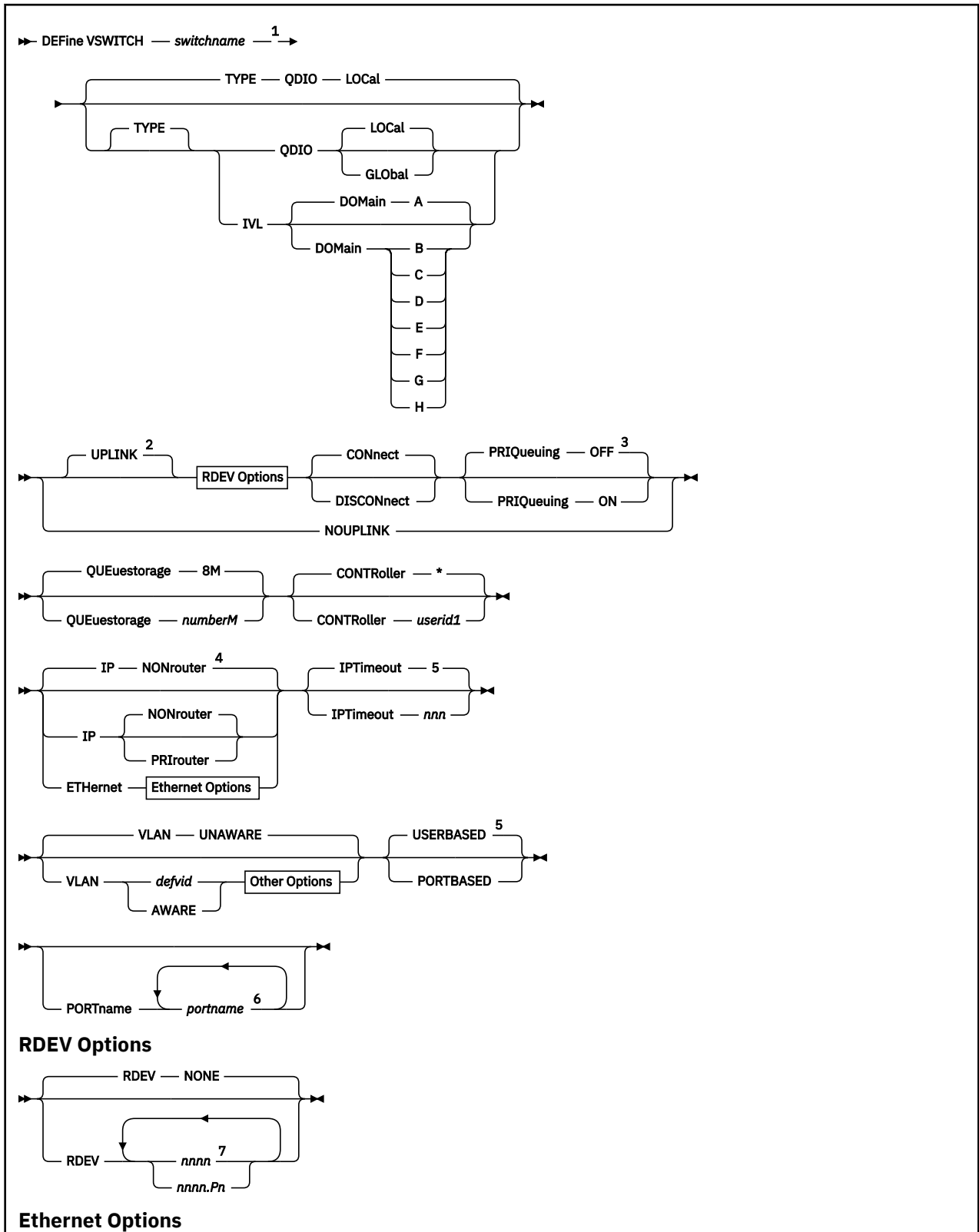
```
DASD vdev DEFINED
```

confirms the creation of the virtual disk in storage.

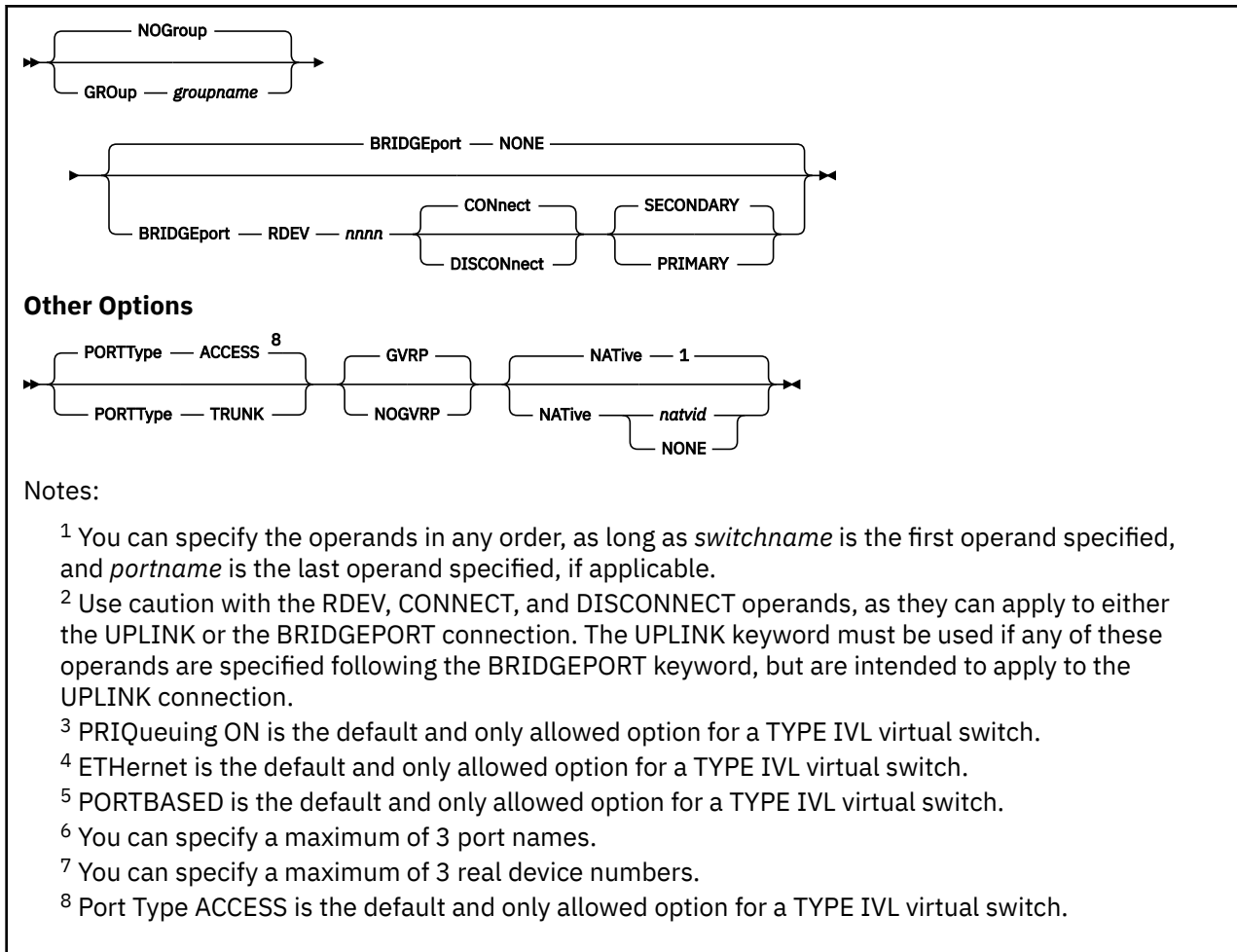
### **Messages**

For the list of general messages for the DEFINE command, see [“Messages” on page 122](#).

# DEFINE VSWITCH







## Authorization

Privilege Class: B

## Purpose

Use the DEFINE VSWITCH command to create a CP system-owned switch (a virtual switch) to which virtual machines can connect. Each switch is identified by a *switchname*. A z/VM user can create the appropriate QDIO network interface card (NIC) and connect it to this switch. For information about related commands, see [“COUPLE” on page 83](#) and [“DEFINE NIC” on page 194](#). For more information about [Virtual Networking Options](#), see *z/VM: Connectivity*.

The SET VSWITCH command can be used to modify attributes of the virtual switch. See [“SET VSWITCH” on page 1757](#) for more information on this command.

You can also define a virtual switch during system initialization using the DEFINE VSWITCH configuration file statement. For more information about the [DEFINE VSWITCH Statement](#), see *z/VM: CP Planning and Administration*.

## Operands

### *switchname*

is the name of the new virtual switch. The *switchname* is a single token (1–8 alphanumeric characters) that identifies this virtual switch for subsequent commands.

### TYPE

specifies the type of virtual switch to be created, specifically the hardware and protocol the virtual switch will emulate.

**QDIO**

defines a simulated Ethernet or IP virtual switch. A QDIO virtual switch creates a network comprised of both simulated QDIO devices residing on the same z/VM system, with real network devices located on an external or physical network. A QDIO type virtual switch can only accept connections from a simulated QDIO adapter. External connectivity to network devices on a physical network is achieved using the RDEV or GROUP keywords.

**IVL**

defines an Inter-VSwitch Link which provides the communication facility to implement an IVL Domain. An IVL domain is a grouping of up to 16 systems running z/VM connected by an IVL LAN segment. All the active members within an IVL domain provide control operations that support the creation and management of shared virtual networking components such as Shared Port Groups. The IVL virtual switch provides communication infrastructure to exchange control information and data necessary to manage global networking objects that can span multiple systems running z/VM. The IVL virtual switch must be defined with external connectivity (using the RDEV or GROUP keywords) and its UPLINK connection must remain operational in order to support global networking objects.

Only one IVL virtual switch may be created on a system running z/VM.

Guest NICs may not be coupled to an IVL virtual switch.

**DOMain [A | B | C | D | E | F | G | H]**

defines the domain to which the IVL virtual switch belongs. See [“19” on page 243](#) for more information about an IVL virtual switch.

**GLobal**

identifies this virtual switch as a member of a global virtual switch. A global virtual switch is a collection of virtual switches that share the same name and the same networking characteristics. This collection of virtual switches spans multiple systems running z/VM but logically operates as a single switch.

Global virtual switches using a shared port group must reside on LPARs in the same CEC.

See [usage note 18](#) for more information about a global virtual switch.

**LOCal**

LOCAL means that the virtual switch is not a member of a global virtual switch.

**UPLINK**

enables and specifies connectivity for a virtual switch UPLINK port. The UPLINK port is a special port that typically is used to connect the virtual switch to a physical switch, essentially bridging the virtual switch's simulated network to a physical network.

**RDEV *nnnn*****RDEV *nnnn.Pn***

is a real device number or a real device number and OSA-Express port number to be used as an UPLINK port to connect the virtual switch to the appropriate model OSA-Express device. The device selected must be compatible to the type of virtual switch created. Connectivity to the physical network will be prevented when the specified device doesn't match the virtual switch's type.

Specify each real device number as a hexadecimal number between X'0001' and X'FFFD'. If specifying a real device and an OSA-Express port number, specify the real device number as a hexadecimal number between X'0001' and X'FFFD' followed by a period (.), the letter 'P' (or 'p') followed by the port number as a hexadecimal number between X'0' and X'F'. For example, to specify port 1 for RDEV 300, specify "300.P1". The value of the port number depends on how many ports the OSA-Express hardware adapter supports. If the port number is not specified, it will default to port 0.

You can specify a maximum of three real device numbers. If you specify more than one device number, each must be separated from the others by at least one blank. When the virtual switch has been defined with the GROUP attribute, any devices identified by the RDEV keyword are used for failover in the event of a real switch failure of the link aggregation group. Failover in this environment will be to a single OSA-Express device connected to a second real switch.

Each real device number represents a trio of devices. For example, specifying **RDEV 111 222 333** means that the first devices, 111-113, are used to provide the connection to the real hardware LAN segment. When issuing the VSWITCH command without the GROUP attribute, if there is a problem with the connection, devices 222-224 are used next to provide the connection, and if those devices fail to connect, devices 333-335 are used. This feature provides dynamic recovery for OSA-Express device failures when issuing the VSWITCH command without the GROUP attribute or for real switch failure when issuing the VSWITCH command with the GROUP attribute. (Failure of an OSA in an aggregated group is automatically handled by the virtual switch; that is, the virtual switch will transfer the data flow to the remaining OSAs in the group.)

RDEV NONE means that the virtual switch should not be connected to the real LAN segment when defined with NOGROUP. When the virtual switch has been defined with GROUP, RDEV NONE means that there is no link aggregation group failover in the event the real switch should fail.

The RDEV operand may not be used to specify device numbers when the virtual switch is configured to use a shared port group. MAC address takeover is managed by the shared port group to maintain connectivity following a failure.

### **CONNECT**

indicates that the currently configured virtual switch UPLINK port must be activated, and traffic may flow through the specified UPLINK ports device(s).

### **DISCONNECT**

indicates that the currently configured virtual switch UPLINK port must not be activated, and that no traffic is to flow through the specified UPLINK ports device(s).

A virtual switch can be functional without a connection to a real LAN segment, and traffic flows only between virtual machines coupled to the virtual switch.

### **PRIQUEUING**

enables or disables guest priority queuing support on all outbound data transmissions from the virtual switch uplink port to an external network. Priority queuing is a capability of the OSA-Express feature where multiple output queues are defined for a single network connection with each queue weighted by a priority on how often it gets serviced by the OSA-Express feature. The highest priority queue is serviced first and more often followed by the next highest priority and so on. No queue is starved and all will get serviced at some point by the OSA-Express feature. See [Virtual Switch Priority Queuing Function](#) in *z/VM: Connectivity* for more information on virtual switch exploitation of priority queuing.

If PRIQUEUING is to be enabled on a virtual switch, then the OSA-Express features configured to the virtual switch's uplink port must be configured by IOCP to enable the feature within the adapter (PQ\_ON). (See ["DEFINE CHPID / PATH"](#) on page 132 for more information.)

### **OFF**

The virtual switch will not exploit priority queuing. A single input queue is established for inbound transmissions from the external network and a single output queue is established for outbound transmissions. All outbound data to the external network is transmitted with equal priority. This is the default for TYPE QDIO virtual switches.

The OFF option is not allowed for a TYPE IVL virtual switch.

### **ON**

The virtual switch will exploit priority queuing. If the customer configured the OSA-Express features used by a virtual switch uplink port for priority queuing via IOCP (PQ\_ON), then z/VM will establish one input queue and four output queues when activating its network connection. This will allow z/VM to transmit data to the external network at four different priorities. CP will use the highest priority queue for control and management traffic. The other three queues (low, normal and high) can be used for virtual NICs' network connections. This is the default for TYPE IVL virtual switches.

For an IVL virtual switch, z/VM will attempt to establish an active network connection with the first OSA device specified. For migration reasons, if the OSA-Express features (used by the IVL virtual switch uplink port) are configured without priority queuing (PQ\_OFF), z/VM will establish an active

## DEFINE VSWITCH

network connection, and force priority queuing off. A warning message will be displayed to inform the customer to configure priority queuing via IOCP.

If the OSA-Express features used by a non-IVL virtual switch uplink port are configured without priority queuing (PQ\_OFF), z/VM will not establish an active network connection, and an error message will be displayed.

The policy used to select which priority a specific datagram is transmitted to the external network is determined by the type of virtual switch.

For an IVL virtual switch, the priority of outbound transmissions is handled by z/VM. IVL management traffic will be queued and transmitted on a high priority queue, and encapsulated production data failover traffic will be sent on a lower priority queue. The default for TYPE IVL virtual switches is PRIQueuing ON so that IVL network management data and encapsulated production data can be prioritized appropriately. Only an OSA-Express feature configured for priority queuing should be used on an IVL virtual switch's uplink port.

For all other virtual switch types, the priority can be set to low, normal or high for all packets sent from a guest NIC's network connection to an external network via the SET VSWITCH command. See the PQUPLINKTX operand on [“SET VSWITCH” on page 1757](#) for more information.

### NOUPLINK

indicates the virtual switch will never have connectivity to a physical network through the UPLINK port. This option removes the UPLINK port from the virtual switch. Once the UPLINK port is removed, it can never be added back to the virtual switch.

Defining a virtual switch UPLINK port with either the RDEV or GROUP operands while also removing the UPLINK port with the NOUPLINK operand will cause the command to fail.

The NOUPLINK option is not allowed for a TYPE IVL virtual switch.

### QUEuestorage *numberM*

indicates the upper limit of the amount of fixed storage CP and Queued Direct I/O Hardware Facility will use for buffers for each OSA-Express data device.

*number* defines the maximum number of megabytes of storage that can be consumed for QDIO queues on a single OSA port. When multiple OSA devices are defined in a link aggregation group, then each OSA port within the group will use the specified amount of storage. Fixed storage is allocated as needed based on network traffic, until the maximum of *numberM* are allocated.

*number* is a number from 1 to 8. 8 is the default value.

### CONTRoller \*

#### CONTRoller *userid1*

identifies the z/VM user ID that controls the OSA-Express device connected at the device number identified by *rdev*. CONTROLLER \* means CP selects from any of the eligible z/VM TCP/IP stacks. See usage note [“3” on page 239](#) for more information about the function of a controller.

If you specify multiple real devices on the RDEV keyword, or through the GROUP keyword, then specify CONTROLLER \*, or allow it to default. The controller functions are then spread across multiple z/VM TCP/IP stacks, providing more flexibility in case of a failure. You can also specify a pool of specific controllers to be chosen from by specifying a list of user IDs after the CONTROLLER keyword with the SET VSWITCH command or the MODIFY VSWITCH statement.

### IP

#### ETHernet

indicates whether the transport for the virtual switch is ETHERNET or IP. The ETHERNET transport type is Data Link (Layer 2) based, where the ETHERNET frame is used as a point of reference for source and destination Media Access Control (MAC) addresses in transporting ETHERNET frames on the LAN. The IP transport type is Network (Layer 3) based, where the IP packet is used as the point of reference for source and destination IP addresses in transporting IP packets on the LAN.

If this option is omitted for a type QDIO virtual switch, then the transport is set to IP NONROUTER. For an IVL virtual switch the ETHERNET setting is used. An IVL virtual switch can function correctly only

at the Data Link (layer 2) level. Specifying anything other than ETHERNET for an IVL virtual switch will cause the command to fail.

#### **NONrouter**

indicates that the OSA-Express device identified by the RDEV keyword will not act as a router to the virtual switch. If a datagram is received at this device for an unknown IP address, the datagram will be discarded. This is the default.

#### **PRIrouter**

indicates that the OSA-Express device identified by the RDEV keyword will act as a primary router to the virtual switch. If a datagram is received at this device for an unknown IP address, the datagram will be passed to the virtual switch.

#### **GRoup *groupname***

indicates that the virtual switch UPLINK port is to be configured to use IEEE 802.3ad Link Aggregation. The *groupname* is a single token (1-8 alphanumeric characters) that identifies the group. Use the SET PORT GROUP command to specify the attributes of the group and the OSA-Express devices that will make up the group. This option can only be specified when the virtual switch has been defined with the ETHERNET transport attribute. **NOGROUP** means that the virtual switch will not use Link Aggregation. The port group must be defined using the SET PORT GROUP command before creating a GLOBAL virtual switch. For a LOCAL virtual switch, if the group specified has not been defined, it will be created with this command.

#### **NOGROUP**

means that the virtual switch will not use Link Aggregation.

#### **BRIDGEport**

configures and specifies connectivity for a virtual switch Bridge Port. The Bridge Port is a special port that is used to connect the virtual switch to a HiperSockets CHPID, essentially bringing the HiperSockets CHPID to the virtual switch's simulated and physical networks.

Configuring a virtual switch Bridge Port is only supported when the Bridge facility is supported by the processor, the virtual switch transport type is ETHERNET, and the virtual switch TYPE is QDIO. Additionally, the ISOLATION feature cannot be used when the virtual switch has a Bridge Port.

Only guest operating systems running in a virtual machine under z/VM and exploiting QDIO Enhanced Buffer State Management (QEBSM) are eligible to be bridged from the HiperSockets CHPID to the virtual switch's simulated and physical networks.

#### **NONE**

specifies that the virtual switch does not have a Bridge Port.

#### **RDEV *nnnn***

is a real device number to be used as a Bridge Port to connect the virtual switch to a HiperSockets CHPID. Only devices that have been configured with the HiperSockets CHPID parameter EXTERNAL\_BRIDGED can be specified. Use these parameters on the DEFINE CHPID command or the appropriate CHPARM in the IOCP (x4 for EXTERNAL\_BRIDGED).

The device selected must be compatible to the type of virtual switch created; for example EXTERNAL\_BRIDGED for a QDIO type virtual switch. Connectivity to the HiperSockets CHPID will be prevented when the specified device does not match the virtual switch's type.

You must specify a single real device number as a hexadecimal number between X'0001' and X'FFFD'. The real device number specified represents a trio of devices. For example, specifying BRIDGEPORT RDEV 508 means the devices, 508-50A, are used to provide the connection to the HiperSockets CHPID.

#### **CONnect**

indicates that the device identified by the RDEV keyword will be immediately activated, allowing the connection to be used as the active or standby Bridge Port.

#### **DISCONnect**

indicates that the device identified by the RDEV keyword will be placed in the inactive state. If this connection is the active Bridge Port connection, another virtual switch with a Bridge Port on the same CHPID in standby state will take over the active Bridge Port connection.

### SECONDARY

indicates that this virtual switch should be assigned a role as a SECONDARY Bridge Port for the HiperSockets CHPID. This is the default. See usage note [“17” on page 242](#) for more information about the SECONDARY Bridge Port role.

### PRIMARY

indicates that this virtual switch must be assigned the role as the PRIMARY Bridge Port for the HiperSockets CHPID. See usage note [“17” on page 242](#) for more information about the PRIMARY Bridge Port role.

### IPTimeout *nnn*

indicates the length of time in minutes that a transient IP address table entry remains in the IP address table for the virtual network. A transient entry is an entry added to the virtual network by one guest on behalf of another. This operand has no meaning for ETHERNET networks.

*nnn* is a number from 1 to 240. 5 minutes is the default value.

### VLAN

used to enable and configure IEEE standard 802.1Q VLAN support for the virtual switch being defined. A VLAN-aware virtual switch provides VLAN controls at the user level (with SET VSWITCH GRANT, SET VSWITCH PORTNUMBER and SET VSWITCH VLANID commands) that cannot be overridden by a guest host. If this option is omitted for a type QDIO virtual switch, then the virtual switch is set to VLAN UNAWARE.

### UNAWARE

indicates the virtual switch does not support IEEE standard 802.1Q. All frames flow within the virtual switch regardless of presence or absence of Virtual Local Area Network (VLAN) tags. Any VLAN tags present in the frames will be ignored within the switch (however the guest hosts may perform VLAN filtering at the virtual NIC level).

### *defvid*

defines the virtual switch as a VLAN-aware switch supporting IEEE standard 802.1Q. The *defvid* defines the default VLAN ID to be assigned to guest ports when no VLAN ID is coded on the SET VSWITCH GRANT VLAN command, MODIFY VSWITCH GRANT VLAN statement, the SET VSWITCH PORTNUMBER command, the SET VSWITCH VLANID command, or through an ESM. It is a number from 1 to 4094.

### AWARE

defines the virtual switch as a VLAN-aware switch supporting IEEE standard 802.1Q without a default VLAN ID. When a virtual switch is specified as VLAN AWARE, one or more VLAN IDs must be assigned to each guest port by either a SET VSWITCH GRANT VLAN command, MODIFY VSWITCH GRANT VLAN statement, the SET VSWITCH PORTNUMBER command, the SET VSWITCH VLANID command, or through an ESM. Failure to assign an explicit VLAN ID causes all untagged frames transmitted from the port to be discarded. In the case of a VLAN-unaware guest using a PORTTYPE ACCESS all outbound frames will be discarded until a VLAN ID is set for the port.

### PORTType ACCESS

defines the default porttype attribute for guests authorized for the virtual switch. For PORTTYPE ACCESS, the guest is unaware of VLAN IDs and sends and receives only untagged traffic.

### PORTType TRUNK

defines the default porttype attribute for guests authorized for the virtual switch. For PORTTYPE TRUNK, the guest is VLAN aware and sends and receives tagged traffic for those VLANs to which the guest is authorized. If the guest is also authorized to the *natvid*, untagged traffic sent or received by the guest is associated with the native VLAN ID (*natvid*) of the virtual switch.

PORTTYPE TRUNK is not allowed for a TYPE IVL virtual switch.

### GVRP

indicates that the VLAN IDs in use on the virtual switch should be registered with GVRP-aware switches on the LAN. This provides dynamic VLAN registration and VLAN registration removal for networking switches. This eliminates the need to manually configure the individual port VLAN assignments.

**NOGVRP**

Do not register VLAN IDs with GVRP-aware switches on the LAN. When NOGVRP is specified VLAN port assignments must be configured manually.

**NATive *natvid*****NATIVE NONE**

the *natvid* option is available only on a VLAN-aware switch. When *natvid* is a number from 1 to 4094, this operand defines the native VLAN ID that is to be associated with untagged frames received and transmitted by the virtual switch. NATIVE NONE causes all untagged frames to be discarded, instead of being delivered with a native VLAN ID. If this option is omitted, 1 is used as the *natvid* for a type QDIO virtual switch.

**USERBASED**

operational differences between USERBASED and PORTBASED VSwitches have been eliminated.

USERBASED specifies that user based rules will be applied when relocating a guest in an SSI. A user-defined port may or may not be preserved over a relocation. Port numbers assigned by z/VM will not be preserved. New port numbers will be assigned on the destination system. If portnumber predictability across an SSI is required, use the PORTBASED option on the VSwitch in each SSI member.

Refer to [What Is the Difference Between a Port Based and User Based Virtual Switch?](#) in *z/VM: Connectivity* for more information.

**PORTBASED**

operational differences between USERBASED and PORTBASED VSwitches have been eliminated.

PORTBASED specifies that port based rules will be applied when relocating a guest in an SSI. User-defined port numbers will be preserved over a relocation. Port numbers assigned by z/VM will not be preserved. In this case a new port number will be allocated on the destination system.

Refer to [What Is the Difference Between a Port Based and User Based Virtual Switch?](#) in *z/VM: Connectivity* for more information.

**PORTname *portname***

is a 1- to 8-character name that identifies the OSA-Express adapter. You can specify a maximum of three port names. Multiple port names are used when different port names are needed for the multiple *rdevs* specified on the RDEV operand. See usage note [“10” on page 241](#) for more information.

**Note:** The PORTNAME operand is ignored when the GROUP operand is used.

**Usage Notes**

1. The DEFINE VSWITCH command creates a virtual switch. The SET VSWITCH command can be used to modify a virtual switch by authorizing users to use the switch. Authorization to a virtual switch may also be provided by an External Security Manager.
2. Accounting is set for the switch based on the default accounting state as set by the SET VMLAN ACCOUNT SYSTEM command or configuration statement. If accounting is turned on after the virtual switch is defined, the virtual switch will need to be redefined for accounting to take effect.
3. A virtual switch's connection to a real hardware LAN segment is not operational until an eligible z/VM TCP/IP stack is selected to be the controller for the OSA-Express device. CP selects an eligible z/VM TCP/IP stack to be the controller by either:
  - If CONTROLLER *userid1* is specified on the DEFINE or SET VSWITCH commands or the DEFINE or MODIFY VSWITCH System Configuration statements, with either a single user ID or a list of user IDs, only those user IDs are selected.
  - If CONTROLLER \* is specified or allowed to default, CP selects from any eligible z/VM TCP/IP stacks.

A z/VM TCP/IP stack becomes eligible when:

## DEFINE VSWITCH

- The TCPIP MODULE running in the controller is at a release level that supports the function required for the virtual switch. It is recommended that the VSWITCH controller be at the same release level as CP, although all supported releases are allowed.
- An IUCV \*VSWITCH statement is included in its CP directory entry.
- The TCP/IP VSWITCH CONTROLLER statement is coded, and has defaulted to be ON or is explicitly set to ON in the TCP/IP configuration file or through an OBEYFILE command.
- The stack has completed initialization.
- The stack has virtual device numbers available for CP to attach the control device.

The virtual device range used by CP is specified in the VSWITCH CONTROLLER TCP/IP configuration statement. If no VDEV range is specified, CP uses the virtual device number (*vdev*) that matches the *rdev* number specified on the DEFINE VSWITCH or SET VSWITCH command. See the [VSWITCH CONTROLLER Statement](#) in *z/VM: TCP/IP Planning and Customization* for more information.

**Note:** Do not code DEVICE and LINK TCP/IP configuration statements for the device. Do not attach the device to a TCP/IP controller virtual machine. These steps are handled by DEFINE VSWITCH processing when a controller is selected.

If an eligible stack is not found, or none of the *rdevs* are operational, you receive a message, and the virtual switch operates in a local LAN environment.

4. The virtual switch supports two modes of operation for data transport in support of both TCP/IP and non-IP based applications. In deciding which mode to deploy for your network, some things to consider about deploying an ETHERNET virtualized LAN segment are:
  - Do your servers or applications need to have their own unique MAC addresses (load balancers)?
  - Do you plan to deploy non-IP based applications on your network (SNA or NetBIOS, for example)?
  - Do you want to build a virtual LAN segment that operates closely to its physical counterpart?

The key attributes of each transport mode with their operational characteristics are as follows:

- **ETHERNET (Layer 2)**

- Supports all applications that deploy ETHERNET (IEEE 802).
- ETHERNET frames are transported on the LAN segment.
- All destinations are identified by MAC address.
- MAC addresses are locally administered by the LAN administrator through z/VM CP commands or configuration statements.
- Each host connection is identified by a single MAC address.
- This is a Link Layer transport, in which all hosts maintain their respective ARP caches.
- VLAN tagging resides within the ETHERNET frames per IEEE 802.1Q specifications.
- When GROUP attribute is specified, the frames can be transported as part of the IEEE 802.3ad standard.
- When a Bridge Port is defined the virtual switch provides physical LAN connectivity for the target HiperSocket channel.

- **IP (Layer 3)**

- Supports IP for TCP/IP applications only.
- IP packets are transported on the LAN segment.
- All destinations are identified by IP addresses.
- IP address assignments are set by the host running in the guest virtual machine.
- Each host may have more than one IP address (multi-homed).
- This is Link Layer independent (that is, no MAC addresses), and ARP processing is offloaded onto the OSE-Express adapter.
- VLAN tagging resides in internal QDIO headers.



- All hosts share the OSA-Express MAC address.

A virtual switch operates in only one mode for a given instance. For example, if the switch is configured as IP, then all communications on the LAN segment must be IP based. The same is also true when the switch is configured in ETHERNET mode. These transport modes affect the method of data transfer for the virtual switch. The other operations of the switch, such as guest authorization, failover, controller configuration, and so on, function the same for both modes.

5. The IP transport type is IPv4 only. In order to support IPv6 through the virtual switch UPLINK port, the ETHERNET transport is required.
6. SET VSWITCH cannot be used to change the type of transport. The virtual switch will need to be redefined.
7. Use the QUERY CONTROLLER command output to find the z/VM TCP/IP stacks that are the virtual switch controllers. Use the QUERY VSWITCH command to display information about the virtual switch.
8. CP manages the devices used to control a virtual switch's connection to a real LAN segment through an OSA-Express device. CP attaches the devices to the z/VM TCP/IP virtual machine. CP also defines devices of type VSWITCH-OSD to the z/VM TCP/IP stack, concatenating *switchname* with *vdev* and "DEV" to form the device name and *switchname* with *vdev* and "LINK" to form the link names. These names appear in the TCP/IP query and trace information.

DEVICE and LINK statements must not be included in the TCP/IP configuration file for these devices.

The port number specified by *nnnn.Pnn* will be used by the TCP/IP stack (controller) when initializing the device. If the port number is not specified, it will default to port 0. If the port number is not supported, initialization of the device will be terminated.

9. Multiple real devices and portnames can be specified on the RDEV and PORTNAME parameters. This feature allows failover to an alternate real device in the event of a failure with the current OSA-Express device or link aggregation group. All real devices specified must be active and connected to the same hardware LAN in order to effectively and dynamically failover to an alternate device. In addition, the alternate devices must be defined on separate CHPIDs. If your OSA-Express device requires a portname, specify one portname for each real device number.
10. When the real device identified by one of the *rdevs* is started using the VSWITCH command with the NOGROUP attribute, TCP/IP assigns the port name as the hardware adapter name. If an adapter name was already assigned by a previous connection, then the port name must be the same as assigned by any other connection in order to share the adapter. This includes sharing the OSA-Express adapter with this logical partition or all other partitions.

The PORTNAME operand is optional. However, some levels of the OSA-Express adapters require that the PORTNAME operand is specified. When such an adapter is in use and the PORTNAME operand is omitted, an error message is displayed during switch initialization.

If the device is already started, you must stop it by issuing the SET VSWITCH *switchname* DISCONNECT command before changing the port name.

11. PRIROUTER is required only when IP forwarding (routing) nodes will be coupled to the switch. Router nodes provide connectivity for their LAN segments (remote nodes) through their switch connection. When Router nodes are deployed, their switch connection must be configured as PRIROUTER. In addition to this, the switch itself must also be configured as PRIROUTER to the OSA-E adapter. This will insure delivery of datagrams destined for LAN segments that are connected through routers coupled to a switch. Only one connection on each OSA-Express card can be designated as PRIROUTER. If the switch is successful in establishing PRIROUTER on the OSA-Express card, no other node (or switch) sharing the same OSA-Express card will be able to act as PRIROUTER. If another connection has already been established as PRIROUTER, the switch will be left with NONROUTER status (which is reflected in the QUERY VSWITCH response).
12. NONROUTER is the default mode for the switch. Every node is directly coupled to the switch and the associated IP destinations are registered with the OSA-Express connection. This is the most efficient way to use the virtual switch. In this mode, packets with an unrecognized IP destination are automatically sent out through the switch connection.

## DEFINE VSWITCH

13. For a VLAN aware virtual switch, you should specify the same native VLAN ID (*natvid*) as specified in your configuration of any physical switches in your network. Most hardware manufacturers use a default of 1.
14. SET VSWITCH cannot be used to change the VLAN awareness attribute. To change the attribute, detach the virtual switch and define it again with the correct attribute.
15. If the virtual switch is defined as VLAN UNAWARE, any attempts to define a VLAN membership will fail. When the CP access list is used, SET or MODIFY VSWITCH GRANT with VLAN membership list fails. In addition, SET or MODIFY VSWITCH VLANID fails and SET or MODIFY VSWITCH PORTNUMBER with VLAN members fails. When an ESM is used, the COUPLE command fails if a VLAN list is returned by the ESM.
16. SET VSWITCH cannot be used to change the USERBASED/PORTBASED attribute. The virtual switch will need to be redefined.
17. A virtual switch's Bridge Port can be configured to take on the role as either the PRIMARY or as a SECONDARY Bridge Port for the HiperSockets CHPID. SECONDARY is the default role assigned when defining a virtual switch with a Bridge Port. A single PRIMARY Bridge Port and up to four SECONDARY Bridge Ports can be defined per HiperSockets CHPID.

A functional virtual switch configured for the PRIMARY role is always selected as the *active* Bridge Port connection. Functional virtual switches configured with the SECONDARY role provide backup (standby) capability in the event of failure on the part of the active Bridge Port connection.

The first virtual switch Bridge Port successfully connecting to the HiperSockets CHPID (whether PRIMARY or SECONDARY) will take on the responsibility as the active Bridge Port connection. When the active Bridge Port connection is a virtual switch with a SECONDARY role, then its responsibility as the active Bridge Port connection will be relinquished to a connection made by a virtual switch requesting the PRIMARY role.

18. A Global virtual switch is a collection of virtual switches that share the same networking characteristics. This collection of virtual switches spans multiple systems running z/VM but logically operates as a single switch.

Virtual switches defined with the same name and the GLOBAL option are said to be members of the same global virtual switch when they reside in systems running z/VM that belong to the same IVL domain. (Definition and activation of an IVL virtual switch allows a system running z/VM to join an IVL domain. See [usage note 20](#) for more information about the IVL virtual switch.)

The following conditions must be fulfilled in order to create or change a global virtual switch member:

- The system's IVL virtual switch must be defined and its UPLINK port connected. In other words, the system must be an active member of an IVL domain.

Creation of Global virtual switches specified in the SYSTEM CONFIG file will be deferred until the IVL virtual switch UPLINK port connects the system to the IVL domain. Message HCP3178I will be displayed for each deferred Global virtual switch.

- If any other virtual switch exists in the IVL domain with the same name the following attributes must match or the DEFINE VSWITCH or SET VSWITCH will fail with message HCP3170E.
  - TYPE QDIO
  - IP or ETHERNET
  - ISOLATION
  - VEPA
  - VLAN AWARE or UNAWARE
  - NATIVE *natvid*
  - USERBASED or PORTBASED
- The following additional restrictions exist if a global virtual switch is configured to use a shared port group:
  - RDEV *device\_addr* is not allowed

19. A virtual switch defined to be TYPE IVL is an Inter-VSwitch Link which provides communication infrastructure to exchange control information and data necessary to manage global networking objects that can span multiple systems running z/VM.

A single IVL virtual switch accommodates all inter-LPAR control traffic for a system. This virtual switch is defined by the system administrator using a CP command or a statement in the System Configuration file. The DEFINE VSWITCH specifies the RDEV or GROUP operand to configure OSA-Express adapters to provide the required connectivity to systems running z/VM in other LPARs so they can all join the same IVL domain.

An IVL domain is a group of systems running z/VM that have each defined an IVL VSwitch with an UPLINK port configured and connected to the same external LAN segment. Configuration of an IVL virtual switch defines the system's IVL domain membership.

These domains allow isolation of global networking traffic through the enumeration of different domain letters A-H. Each is assigned a separate, reserved multicast MAC address. For example, communications for default Domain A is assigned to 03-FF-FF-FF-FF-01. (MAC address prefix 03-FF-FF is reserved for IVL communications for all systems running z/VM.)

The VLAN associated with a VLAN-aware IVL virtual switch can be modified using the IVLPORT VLAN operand on the SET VSWITCH command. Up to 8 IVL domains can be defined per VLAN.

Note that up to 16 systems can be members of the same IVL domain.

When a global virtual switch is in use the IVL virtual switch UPLINK port must remain operational in order to support full function for a global virtual switch that spans multiple systems.

See [IVL \(Inter-VSwitch Link\) Overview](#) in *z/VM: Connectivity* for more information.

## Examples

1. To define a switch named BIGANG that connects to a real LAN through device fd00, specify the following:

```
define vswitch bigang rdev fd00
```

2. To define a switch named ETH1 that will use Link Aggregation, specify the following commands:

```
set port group eth1grp join b c d
define vswitch eth1 ethernet group eth1grp
```

3. To define a switch named LINPROD that uses OSA-Express hardware ports, specify the following:

```
define vswitch linprof rdev 9c00.p0 9d00.p1 9e00
```

In the above example, device 9e00 will connect to port 0 on the OSA-Express.

4. To define a switch named ETH0 that connects to a real LAN through device 1e1b and to a HiperSockets CHPID through device 7000, specify the following:

```
define vswitch eth0 type qdio rdev 1e1b ethernet bridgeport rdev 7000
```

## Responses

Response 1:

```
VSWITCH SYSTEM switchname is created
```

This is the normal response for this command.

## Messages

For the list of general messages for the DEFINE command, see [“Messages”](#) on page 122.

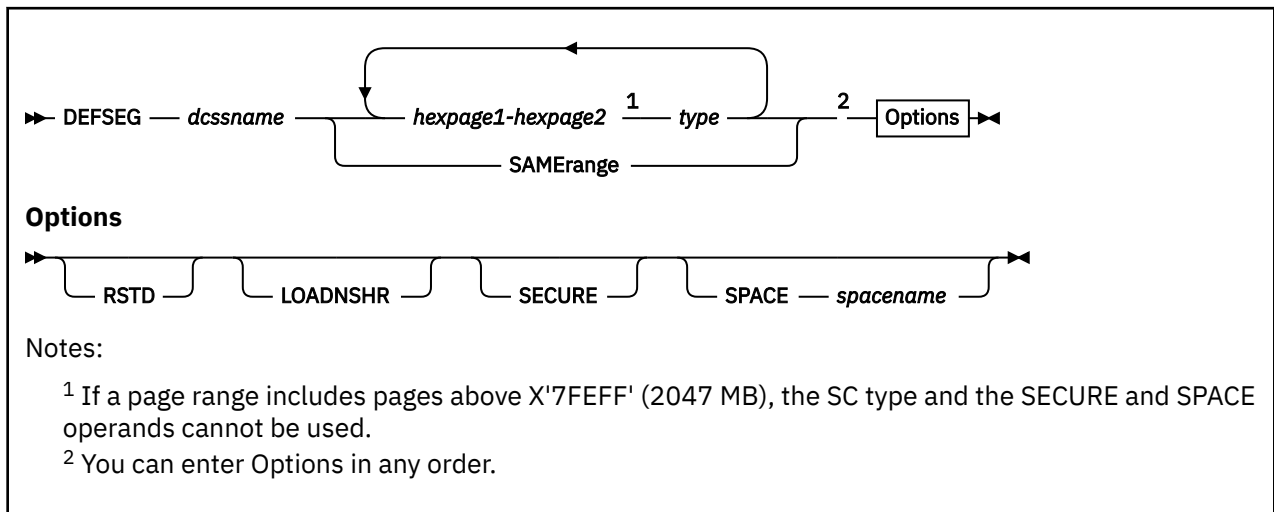
Messages:

## DEFINE VSWITCH

- HCP013E Conflicting option - *option*
- HCP025E Storage missing or invalid
- HCP2768E Missing switchname
- HCP2782E VSWITCH SYSTEM *switchnm* not created; a detach is in progress for a VSWITCH with that name
- HCP2782E VSWITCH SYSTEM *switchnm* not created; this name is already in use
- HCP2782E VSWITCH SYSTEM *switchnm* is not created; an IVL virtual switch already exists.
- HCP2784E VSWITCH *switchnm* not created; VMLAN limit has been reached
- HCP2838I Device *device* specified for VSWITCH *vswitchname* BRIDGEPORT is not a bridge-capable HiperSockets device.
- HCP2838I Device *device* specified for VSWITCH *vswitchname* is not compatible with the VSWITCH type.
- HCP2838I Device *device* specified for VSWITCH *vswitchname* is not configured for priority queuing.
- HCP2846E The [PORTTYPE | GVRP | NATIVE] option is not allowed for a VLAN UNAWARE virtual switch
- HCP2853E VSWITCH SYSTEM *switchname* not created; device limit exceeded.
- HCP2854E VSWITCH SYSTEM *switchname* not created; No MACID is available.
- HCP2854E VSWITCH SYSTEM *switchname* not created; SSI cluster mode is not stable.
- HCP2856E VSWITCH SYSTEM *switchname* not created; port group *groupname* is already in use.
- HCP3001E Device *device* is defined for both port group *groupname* and VSWITCH SYSTEM *switchname*.
- HCP3045E The Bridge facility is not supported by this processor.
- HCP3160E Virtual switch *switchnm* cannot be configured with the RDEV operand specifying back up devices.
- HCP3161E [ DEFINE VSWITCH | SET VSWITCH ] is not performed; a port group defined with the SHARED option can be used only with a virtual switch defined with the GLOBAL option.
- HCP3167E [ DEFINE VSWITCH *switchnm* | SET VSWITCH *switchnm* | SET PORT GROUP *groupname* ] failed. The IVL virtual switch does not exist or its UPLINK port is not connected.
- HCP3170E VSWITCH *systemname.switchname* [is not created | is not changed]. The virtual switch defined on *systemname* [ conflicts with the [ IP | ETHERNET | ISOLATION | VEPA | VLAN AWARE | VLAN UNAWARE | NATIVE VLAN | USERBASED | PORTBASED ] attribute ].
- HCP3170E VSWITCH *systemname.switchname* [is not created | is not changed]. An internal error occurred.
- HCP3170E VSWITCH *systemname.switchname* [is not created | is not changed]. The virtual switch defined on *systemname* [ is currently not responding | is currently in synchronization ].
- HCP3173E Device *device* is reserved for system use.
- HCP3178E [ VSWITCH SYSTEM *switchnm* | PORT GROUP *groupname* ] creation is deferred. The IVL virtual switch does not exist or its UPLINK port is not connected.
- HCP3186E [ DEFINE VSWITCH GROUP | SET PORT GROUP | SET VSWITCH GROUP ] for shared port group *groupname* not performed; devices on OSA PCHID *pchid* do not exist or are not available.
- HCP3194I Creating [ VSWITCH *switchnm* | PORT GROUP *groupname* ] now that the IVL Virtual switch is connected.
- HCP3196E VSWITCH *vswname* is not created; the port group must be defined before creating the associated GLOBAL vswitch.
- HCP3199E [DEFINE VSWITCH | SET VSWITCH] not performed; uplink port group *portgroup* on this system is not equivalent to other IVL members sharing the same port group.
- HCP3221E VSWITCH not created; usage of shared port group *groupname* would exceed the maximum number of virtual switches allowed by a single system sharing the same port group.
- HCP3230I Priority Queuing is not enabled for device *device* for VSWITCH SYSTEM *switchname*.

- HCP3230I Priority Queuing is not enabled for VSWITCH SYSTEM *switchname*. No eligible controllers support priority queuing.
- HCP6706E Invalid groupname - *groupname*
- HCP6706E Invalid portname - *portname*
- HCP6706E Invalid switchname - *switchname*

## DEFSEG



### Authorization

Privilege Class: E

### Purpose

Use DEFSEG to define a skeleton system data file (class S, SDF) for a discontinuous saved segment (DCSS). A DCSS has the following attributes:

- It can be loaded at an address above a virtual machine's defined storage size.
- It can include page ranges that are not contiguous.

By using the SPACE operand, you can define special types of DCSSs called segment spaces and member saved segments. Therefore the terms *discontinuous saved segment* and *DCSS* generally refer to saved segments which are not segment spaces or member saved segments.

The skeleton file consists of a descriptor page containing all information necessary to preserve the saved segment with the SAVESEG command. You cannot access the saved segment until it has been saved.

### Operands

#### *dcssname*

is the name of the saved segment (a 1- to 8-character alphanumeric string). When the SPACE operand is used, *dcssname* is known as the member name and must be different from the *spacename* name.

#### *hexpage1-hexpage2*

is a range of pages to be saved. *hexpage1* and *hexpage2* are the numbers of available pages. When they are the same, only that page is saved. The range may be contained in a single 1 MB segment, or it may comprise a contiguous area in two or more segments. You may use multiple page ranges to define a 1 MB segment. However, the page descriptor codes of all page ranges in the same segment must have the same shared or exclusive attribute. (See the description of the *type* parameter for information about descriptor codes.)

Define only those pages whose data must be saved. Any page or pages in a segment not explicitly defined in a page range on the DEFSEG command defaults to the "no data saved" attribute. If other pages in the segment have been defined with the shared attribute (page descriptor codes SR, SW, SN and SC), pages not specified on the command line become shared read-only pages whose data are not saved. No page descriptor code corresponds to this state; it has the same attributes as the state defined by page descriptor code SC, except that CP does not write into these pages.

If other pages in the segment have been defined with the exclusive attribute (page descriptor codes ER, EW, and EN), any pages not specified on the command line become exclusive read/write pages whose data are not saved. This corresponds to page descriptor code EN.

The page ranges of a member saved segment cannot overlap with the specified page ranges of another member saved segment associated with the same segment space. Areas not specified (defaulted to "no data saved") may be taken by another member saved segment. If the member uses these pages but doesn't need the data from the pages saved, the range with the EN or SN attribute type must be specified. Also, this member saved segment's range descriptor code cannot conflict with those of an existing member that has ranges defined in the same segment of storage.

The page number is a hexadecimal value less than or equal to X'7FFFFFFF' (512 GB). The following list shows examples of how hexadecimal storage addresses translate into hexadecimal page numbers:

<b>Storage Address</b>	<b>Page Number</b>
0000000xxx	0
0000001xxx	1
0000022xxx	22
0000333xxx	333
0004444xxx	4444
007F800xxx	7F800
0080000xxx	80000
7F8000Fxxx	7F8000F
0100000xxx	100000

If a page range includes pages above X'7FEFF' (2047 MB), the SC page range descriptor and the SPACE and SECURE operands cannot be used on the same DEFSEG command.

The maximum size of a saved segment is 2047 MB.

### **type**

refers to the page descriptor code of the page range in the saved segment. It indicates the type of virtual machine access permitted to pages in the range.

The first character is either E for exclusive access or S for shared access. Sharing of storage between virtual machines is based on 1 MB segments, so all 256 pages in any one segment must be either exclusive or shared.

The second character of the page descriptor code defines the storage protection characteristics of the page. These character designators are:

#### **R**

Indicates that page protection is used to make the page range read-only.

#### **W**

Means that the page is not protected, and users have read/write access.

#### **N**

Stands for "read/write, no data saved," meaning that the page is treated as a new page when the user refers to it. The contents of, and storage keys for, "no data saved" pages are not saved by the SAVESEG command.

#### **C**

Stands for "CP-writable, virtual machine read-only, no data saved," meaning virtual machines have read-only access to these pages, but CP services may change the data in the pages.

The contents of storage keys for "no data saved" pages are not saved by the SAVESEG command.

Valid page descriptor codes are:

**Code****Meaning****EW**

Exclusive read/write access.

**EN**

Exclusive read/write access, no data saved.

**ER**

Exclusive read-only access.

**SW**

Shared read/write access.

**SN**

Shared read/write access, no data saved.

**SR**

Shared read-only access.

**SC**

CP writable pages, shared read-only access by virtual machine, no data saved.

**Note:** The SC page descriptor code cannot be used with the LOADNSHR operand.

The user must define all pages in segment zero in ranges with an exclusive page descriptor code (EW, ER, or EN). The system rejects the DEFSEG command if the user defines any pages in segment zero with a shared access page descriptor code.

**SAMERange**

specifies that this member saved segment definition is the same as one specified by a previous DEFSEG. This operand is mutually exclusive with the *hexpage1-hexpage2* operands. When you specify SAMERANGE, you also must specify SPACE.

You do not have to specify SAVESEG if you use DEFSEG with the SAMERANGE operand for a class A member saved segment. However, if the member saved segment you are referring to is a class S file, you must still use SAVESEG to save the file.

SAMERANGE makes it easier to create overlaying segment spaces by using the same member for both segment spaces.

If you use this operand in the same DEFSEG command with a name in *spacename* that does not exist, the member that you specify must be a class S (skeleton) file.

You must have either a class S or class A file with the same *dcssname* when you use this operand. If you have both class A and class S files, the command uses the information in the class S file.

**RSTD**

indicates a restricted saved segment. The user must have a corresponding NAMESAVE directory statement to access this saved segment. If any member of a segment space is defined with the restricted operand (that is, the SPACE operand is used), the entire segment space is then restricted. This means that the restricted *spacename* is that which must be used on the NAMESAVE directory statement. If you didn't specify SPACE, then *dcssname* is restricted. If you use the RSTD operand, make sure that you use the NAMESAVE directory statement to allow authorization.

If you used the SAMERANGE operand, you can use RSTD. This affects the authorization of the segment space that you specified in the DEFSEG definition. The name specified in *spacename* must be used on the NAMESAVE directory statement.

The RSTD operand cannot be used with the LOADNSHR operand.

**LOADNSHR**

indicates that any user may load a nonshared copy of the saved segment. No NAMESAVE directory authorization is required. If any member saved segment is defined with LOADNSHR, a nonshared copy of the space itself or any member may be loaded by any user.



The LOADNSHR operand cannot be used with the RSTD operand. It also cannot be used with the SC page range descriptor code.

### SECURE

indicates that the user ID that created the saved segment will be the only user ID allowed to use SPXTAPE DUMP or SPXTAPE LOAD for that file.

### SPACE

specifies the definition of a segment space. Use this operand to make the *dcssname* that you are defining a member of this segment space.

#### *spacename*

is the name of the segment space (a 1- to 8-character alphanumeric string). You must specify this if you use the SPACE operand. The *spacename* must be different from any *dcssname* known to CP.

## Usage Notes

1. CP can process only one DEFSEG or SAVESEG command at a time. If CP is already processing a DEFSEG or SAVESEG command, processing of a subsequent DEFSEG or SAVESEG command is delayed. The SAVESEG command writes all page ranges, except those defined as EN or SN, to the associated system data file. The amount of time it takes for the command to complete is directly proportional to the amount of data to be written.
2. Defining a saved segment at a high storage address that will be shared by many virtual machines might affect real storage availability. For each virtual machine, CP creates dynamic address translation (DAT) tables to reference the virtual machine storage. DAT tables include page tables, segment tables, and higher level (region) tables.

CP keeps the page tables in page management blocks (PGMBKs). Each 8 KB PGMBK references 1 MB of virtual machine storage. For shared page ranges within a saved segment that is loaded shared, the associated segment table entries will point to the same page tables. However, for a saved segment that is loaded nonshared, or for exclusive page ranges within a saved segment, unique page tables are created for each user. PGMBKs might be pageable; as such, their impact on real storage depends on how frequently the MBs of storage they reference are used.

Segment tables and region tables are allocated from host real storage and are not pageable:

- To reference the page tables for a primary address space or data space up to 2 GB, 1 - 4 contiguous frames are allocated for the segment table, one frame for each 512 MB of storage.
- For a primary address space larger than 2 GB, multiple segment tables are created, plus one or more region tables to reference the segment tables. Each region table occupies 1 - 4 contiguous frames. If needed, multiple levels of region tables are created.

Because CP dynamically expands the size of a virtual machine to incorporate a saved segment loaded at an address outside the virtual machine, the DAT tables for the virtual machine also expand. However, addresses between the top of the virtual machine and the bottom of the saved segment are not addressable by the virtual machine.

3. Pages either defined as, or defaulted to, the no data saved attribute are initialized to a page of zeros when first referred to. If the pages are in an exclusive saved segment, changes are seen only by the one user. If the pages are in a shared saved segment, any changes previously made by any sharing user (if allowed) are visible to all other users sharing that same page.
4. Storage keys are not saved for pages defined as, or defaulted to, the "no data saved" attribute. When first referred to, they have keys of zero.
5. A segment space has a range that may span one or more physical MB boundaries. The beginning address of a segment space is rounded down to the nearest MB boundary of the member having the lowest page value. The ending address of a segment space is rounded up to the last address of the last page within the MB occupied by the member having the highest page value.
6. Each member saved segment has a range that may span one or more pages. The beginning address of a member saved segment is determined by its lowest page value. The ending address is determined by the last address of the highest page value.

7. A segment space can include up to 64 member saved segments. A member saved segment can belong to up to 64 segment spaces.
8. Segment zero must always have the exclusive access attribute "Ex". If a shared ("Sx") page descriptor code is used for any pages in segment zero, the command is rejected.
9. Saved segments are loaded through DIAGNOSE code X'64'. You can also use the CMS SEGMENT command or macro, which call DIAGNOSE code X'64'. However, CMS does not support saved segments above 2047 MB. A DCSS that contains pages above 2047 MB can be loaded only through DIAGNOSE code X'64', using a 64-bit subcode.
10. A CP DCSS (defined as type SC) contains CP-writable pages, to which virtual machine users have just read-only access. CP writes data into these pages, which may then be referred to by virtual machines. Serialization of data in the CP-writable pages, if desired, must be arranged between the CP function providing the data and the virtual machines referring to it. A CP DCSS cannot be loaded in nonshared (exclusive) mode by DIAGNOSE X'64'.
11. Only one skeleton (class S) file can exist with a specific name. If one already exists and a second DEFSEG command is issued specifying the same name, the command is rejected.
12. Each segment name may be used for only one of the following: a segment space, a member saved segment, or a DCSS.
13. Defining members that use undefined areas within other members of the same segment space is not recommended. However, in support of multiple products requiring both shared and exclusive segment types and to minimize storage use, definition of members with interleaved ranges is supported.
14. You can use the SAMERANGE operand only with the SPACE operand.  
**Note:** You *cannot* use the SAMERANGE operand with a *dcssname* containing SW, SN, EW, or EN page types.
15. A DEFSEG with a SAMERANGE operand can be the first definition of a segment space only if the specified member is a class S file.
16. A DEFSEG with a SAMERANGE operand does not need a corresponding SAVESEG if the member saved segment is already saved. If the member saved segment is not saved, you need to use the SAVESEG command.
17. A member of a segment space cannot overlap any of the ranges specified for an existing member within the same segment space. Also, the members cannot have conflicting descriptor codes for ranges defined within the same segment of storage.
18. You cannot specify a DEFSEG with the SAMERANGE operand for a segment space that already has a member saved segment with that name.
19. Putting a member name in a directory statement has no effect; use the segment space name.
20. Processing of a DEFSEG command might be delayed if DIAGNOSE code X'64' is concurrently performing the initial load of a large DCSS.
21. If a new version of a NSS or DCSS is created and a previous version (with the same filename) exists and had a reserved storage setting (as established by the SET RESERVED command), the reserved storage setting is not inherited by the new version of the NSS or DCSS. If you want reserved storage for the new version, you must explicitly enter the SET RESERVED command for the new NSS or DCSS after it is saved using SAVESYS or SAVESEG.

#### Security Notes:

1. When RSTD is specified (indicating a restricted saved segment), only those users whose directory entries contain a NAMESAVE statement specifying the name of the restricted system data file are allowed to load this saved segment. If RSTD was specified **without** the SPACE operand, the directory entries must contain a NAMESAVE statement specifying the same DCSS name as *dcssname*. If RSTD was specified **with** the SPACE operand, the directory entries must contain a NAMESAVE statement specifying the segment space name as *spacename*. This allows an installation to ensure data integrity by restricting access to saved segments that contain shared writable storage (defined with the SW

and SN page descriptor codes) or to CP DCSSs (defined with the SC page descriptor code), which may contain sensitive data.

- Virtual machines that share read/write pages, defined with the SW and SN page descriptor codes, are responsible for the integrity of the shared data. The operating system that uses these shared writable pages should ensure that these virtual machines are properly protected from one another.

## Examples

```
defseg userseg 100-1ff sr rstd
defseg seg1 1000-1fff en 2000-2fff sr 3000-3fff er
defseg ppa 700-721 sr space group1
defseg ppb 722-7A1 sr space group1
defseg ppc 7A2-7F0 sr space group1
defseg ppx 700-721 sr rstd space group1a
defseg ppb same space group1a
defseg ppc same space group1a
defseg ppd 700-740 sr space group3
defseg ppe 741-779 sr space group3
defseg ppe 7A0-7FF sr space group3
defseg seg5 800-8FF sr loadnshr
defseg ppg 700-738 sr space group4 loadnshr
defseg ppe same space group4
```

For more explanations and examples, see [z/VM: Saved Segments Planning and Administration](#).

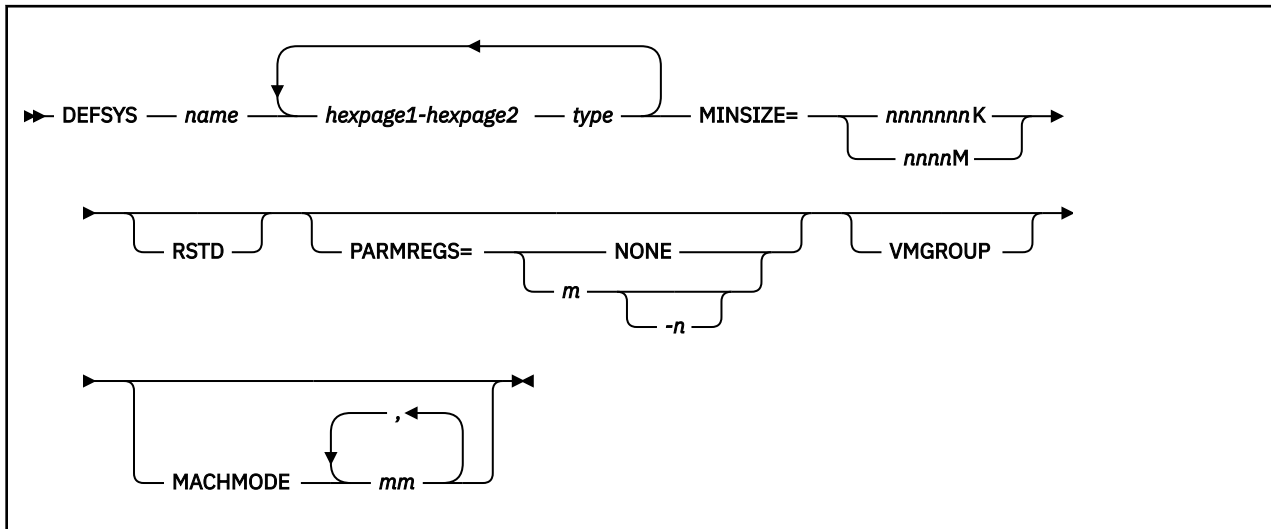
## Messages

- HCP002E Invalid operand - *operand*
- HCP263E Too many operands were supplied for this command.
- HCP299E The saved segment *name* could not be defined because a class S file with the same name already exists.
- HCP339E The command processing cannot complete because the page descriptor code *invalid-code* for the page range *nnnnn-nnnnn* should be exclusive.
- HCP340E The size of a DCSS is larger than the maximum size 2047M.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP427E *type {rdev|vdev|name[.userid]}* system spool space full; file {closed|held|purged}
- HCP435E DEFSEG command processing cannot complete because a paging I/O error occurred while writing a page.
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP440I Saved segment *name* was successfully saved in fileid *fileid*.
- HCP445E The saved segment *name* could not be defined due to an unexpected error.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP847E Maximum system spool file limit exceeded
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*
- HCP1353E An invalid {name *name*|page range *nnnnn-nnnnn*} was detected in the DEFSEG command.
- HCP1354E An invalid storage page descriptor code *invalid-code* was detected for page range *nnnnn-nnnnn*.
- HCP1355E Segment number *nnn* was defined with both shared and exclusive page descriptor codes.
- HCP1356E *nnnnn-nnnnn* is a conflicting page range. Please check for overlaps and reissue the command.
- HCP1360E Segment *name* was not defined because a saved segment with this name already exists.
- HCP1361E The SAMERANGE operand was used with an invalid or missing SPACE operand.

## DEFSEG

- HCP1362E *name* is not a segment space and cannot have members added.
- HCP1363E The specified segment space *name* already has the maximum number of members.
- HCP1365E *name1 spoolid* has an entry for *name2* which was not found, [purge *name1* | purge *name1* or define *name3*|*name1* purged].
- HCP1366E Member saved segment *name* cannot be associated with another segment space.
- HCP1369E *name* is already defined in segment space *name*.
- HCP1371E *name* is not defined as a member saved segment.
- HCP8161E CONFLICT BETWEEN OPERANDS *operand* AND *operand*

## DEFSYS



### Authorization

Privilege Class: E

### Purpose

Use DEFSYS to define a skeleton system data file (Class S, SDF) for a named saved system (NSS).

The skeleton file consists of a descriptor page that contains all information that is necessary to preserve the NSS with the SAVESYS command.

**Note:** Users cannot access the NSS until it has been saved.

### Operands

#### *name*

is the name of the NSS (a 1- to 8-character alphanumeric string).

**Important:** If you specify a name that is the hexadecimal number of a device, that device is IPLed instead of the NSS.

LOADDEV and DUMPDEV are reserved keywords and cannot be used as NSS names.

#### *hexpage1-hexpage2*

is a range of pages to be saved. *hexpage1* and *hexpage2* are the page numbers of available pages. When they are the same, only that page is saved. The range can be contained in a single 1 MB segment, or it can comprise a contiguous area in two or more segments. You can use multiple page ranges to define a 1 MB segment. However, the page descriptor codes of all page ranges in the same segment must have the same shared or exclusive attribute. (See the description of the *type* operand for information about page descriptor codes.)

Define only those pages whose data must be saved. Any page or pages in a segment that is not explicitly defined in a page range on the DEFSYS command defaults to the "no data saved" attribute. If other pages in the segment were defined with the shared attribute (page descriptor codes SR, SW, SN and SC), pages that are not specified on the command line become shared read-only pages whose data is not saved. No page descriptor codes correspond to this state; it has the same attributes as the state defined by page descriptor code SC except that CP does not write into these pages.

If other pages in the segment were defined with the exclusive attribute (page descriptor codes ER, EW and EN), any pages not specified on the command line become exclusive read/write pages whose data is not saved. This corresponds to page descriptor code EN.

The page number is a hexadecimal value less than or equal to X'7FEFF' (2047 MB). The following list shows examples of how hexadecimal storage addresses translate into hexadecimal page numbers:

Storage Address	Page Number
00000xxx	0
00001xxx	1
00022xxx	22
00333xxx	333
04444xxx	4444
3E6FFxxx	3E6FF
7F800xxx	7F800

**type**

refers to the 2-character page descriptor code of the page range in the NSS. It indicates the type of virtual machine access that is permitted to pages in the range.

The first character is either E for exclusive access or S for shared access. Sharing of storage between virtual machines is based on 1 MB segments, so that all 256 pages in any one segment must be either exclusive or shared.

The second character of the page descriptor code defines the storage protection characteristics of the page. These character designators are:

**R**

Indicates that page protection is used to make the page range read-only.

**W**

Means that the page is not protected, and users have read/write access.

**N**

Stands for "read/write, no data saved," meaning that the page is treated as a new page when the user refers to it. The contents of, and storage keys for, no-data-saved pages are not saved by the SAVESYS command.

**C**

Stands for "CP-writable, virtual machine read-only, no data saved," meaning virtual machines have read-only access to these pages, but CP services can change the data in the pages.

The contents of and storage keys for no-data-saved pages are not saved by the SAVESYS command.

Valid page descriptor codes are:

**Code**

**Meaning**

**EW**

Exclusive read/write access.

**EN**

Exclusive read/write access, no data saved.

**ER**

Exclusive read-only access.

**SW**

Shared read/write access.

**SN**

Shared read/write access, no data saved.

**SR**

Shared read-only access.

**SC**

CP-writable pages, shared read-only access by virtual machine, no data saved.

Any page ranges defined in segment zero must have an exclusive page descriptor code (EW, ER, or EN). If any pages in segment zero are defined with a shared access page descriptor code, the command is rejected.

**MINSIZE=nnnnnnnK****MINSIZE=nnnnM**

indicates the minimum storage size of the virtual machine into which the NSS can be loaded. The operand *nnnnnnnK* is the storage size in decimal kilobytes; *nnnnM* is the storage size in decimal megabytes.

**Options****RSTD**

indicates a restricted NSS. The user must have a corresponding NAMESAVE directory statement to access this NSS.

**PARMREGS=*m*****PARMREGS=*m-n*****PARMREGS=NONE**

specifies a range of general purpose registers to contain the IPL parameter string when the NSS receives control from the control program.

The operands *m* and *n* are decimal values from 0 to 15. If you specify two registers, *n* must be greater than or equal to *m*. If only one register is required, you can specify PARMREGS=*m*.

NONE specifies that no parameters can be passed to this NSS.

**Note:** If you want to use the AUTOOCR, BATCH, NOSPROF, INSTSEG, or SAVESYS parameters of the IPL command, you must have **PARMREGS=0-15** in the DEFSYS command for the CMS NSS.

**VMGROUP**

specifies that a user who IPLs this NSS becomes a member of a virtual machine group that is identified by the name of the NSS.

**MACHMODE *mm***

specifies the set of virtual machine modes in which this NSS can be saved and IPLed.

The operand *mm* is a list of one or more virtual machine modes, separated by commas with no intervening blanks. Each entry in this list is one of the virtual machine mode designators ESA, XA, XC, or Z. ESA designates ESA/390 mode, although a guest can switch the virtual machine to z/Architecture mode. XA is supported for compatibility and is functionally equivalent to ESA. XC designates ESA/XC mode, although a guest can switch the virtual machine to z/XC mode. Z designates a guest that runs only in z/Architecture mode.

If more than one machine mode is specified with this operand, the NSS is known as a multimode NSS. Special considerations apply regarding the information that is saved by the SAVESYS command and restored by the IPL command for a multimode NSS; see the description for the SAVESYS command for these special considerations. However, for purposes of determining if the NSS is multimode or not, the designations XA and ESA are considered synonyms. If both are specified, the pair is treated as a single machine mode specification rather than two. That is, specifying 'MACHMODE XA,ESA' does not result in the NSS being considered a multimode NSS.

**Usage Notes**

1. CP can process only one DEFSYS or SAVESYS command at a time. If CP is already processing a DEFSYS or SAVESYS command, processing of a subsequent DEFSYS or SAVESYS command is delayed. The SAVESYS command writes all page ranges, except those defined as EN or SN, to the

associated system data file. The amount of time it takes for the command to complete is directly proportional to the amount of data to be written.

2. Defining an NSS at a high storage address that will be shared by many virtual machines might affect real storage availability. For each virtual machine, CP creates dynamic address translation (DAT) tables to reference the virtual machine storage. DAT tables include page tables, segment tables, and higher level (region) tables.

CP keeps the page tables in page management blocks (PGMBKs). Each 8 KB PGMBK references 1 MB of virtual machine storage. For shared page ranges within an NSS, the associated segment table entries will point to the same page tables. However, for exclusive page ranges within an NSS, unique page tables are created for each user. PGMBKs might be pageable; as such, their impact on real storage depends on how frequently the MBs of storage they reference are used.

Segment tables and region tables are allocated from host real storage and are not pageable:

- To reference the page tables for a primary address space or data space up to 2 GB, 1 - 4 contiguous frames are allocated for the segment table, one frame for each 512 MB of storage.
- For a primary address space larger than 2 GB, multiple segment tables are created, plus one or more region tables to reference the segment tables. Each region table occupies 1 - 4 contiguous frames. If needed, multiple levels of region tables are created.

Because CP dynamically expands the size of a virtual machine to incorporate an NSS loaded at an address outside the virtual machine, the segment table for the virtual machine might also expand.

3. Page zero should always be saved for an operating system. If it is not saved, certain information might be lost. IBM recommends that page zero is saved with the exclusive write (EW) page descriptor code.
4. Segment zero should always have the exclusive access attribute (EW, EN, or ER). If a shared page descriptor code (SW, SN, SR, or SC) is used for any pages in segment zero, the command is rejected.
5. Pages that are either defined as, or defaulted to, the no-data-saved attribute are initialized to a page of zeros when first referred to. If the pages are in an exclusive segment, changes are seen only by the one user. If the pages are in a shared segment, any changes that are previously made by any sharing user are visible to all other users sharing that same page.
6. A CP NSS (SC mode) contains CP-writable pages, to which virtual machine users have just read-only access. CP writes data into these pages, which can then be referred to by virtual machines. Serialization of data in the CP-writable pages can be arranged between the CP function that provides the data and the virtual machines that reference the data.
7. NSSs are loaded by using the IPL command. The use of the PARMREGS values is described with the IPL command.
8. When the user refers to any pages either defined as, or defaulted to, no-data-saved, the user gets a new page. A new page is assigned as a full frame of zeros, just as any page that does not have a copy that is backed on auxiliary storage.
9. Storage keys are not saved for pages that are either defined as, or defaulted to, no-data-saved. When these pages are first referred to, they have keys of zero. The virtual machine can set the keys for writable storage.
10. Each segment has an associated address range of 1 MB, regardless of the number of pages specified.
11. The maximum amount of storage available to a virtual machine is 2047 MB. To specify this value for MINSIZE, enter either '2047M' or '2096128K'.
12. If you do not specify the MACHMODE operand, the NSS can be saved (through the SAVESYS command) regardless of the saver's MACHINE value. However, the NSS can be IPLed only when the IPLer has the same MACHINE value as the saver.

If you specify the MACHMODE operand, the NSS can be saved or IPLed only when the SAVESYS or IPL command is issued by a guest whose MACHINE value is specified on the MACHMODE operand.

13. Virtual machines that are designated as XA virtual machines (for example, by SET MACHINE XA) and virtual machines that are designated as ESA virtual machines are equivalent. Both designations cause



the virtual machines to operate as ESA virtual machines. The virtual machine mode designations XA and ESA can be considered synonyms. Therefore:

- NSSs that are defined by specifying MACHMODE ESA can also be saved when a virtual machine has been designated as an XA virtual machine.
  - NSSs that are defined by specifying MACHMODE XA can also be saved when a virtual machine has been designated as an ESA virtual machine.
  - An NSS that can be IPLed when a virtual machine has been defined as an ESA virtual machine can also be IPLed when a virtual machine has been defined as an XA virtual machine.
  - An NSS that can be IPLed when a virtual machine has been defined as an XA virtual machine can also be IPLed when a virtual machine has been defined as an ESA virtual machine.
14. Processing of a DEFSYS command might be delayed if DIAGNOSE code X'64' is concurrently performing the initial load of a large DCSS.
  15. The NSS name "\_0\_\_\_1\_\_" is reserved for IBM use.
  16. The reserved storage setting is not inherited by the new version of the NSS or DCSS if the following conditions are true:
    - A new version of a NSS or DCSS is created.
    - A previous version (with the same filename) exists.
    - The previous version had a reserved storage setting (as established by the SET RESERVED command).

If you want reserved storage for the new version, you must explicitly enter the SET RESERVED command for the new NSS or DCSS after it is saved by using SAVESYS or SAVESEG.

#### Security Notes:

1. When RSTD is specified, only those users whose directory entries contain a NAMESAVE statement that specifies the same *name* as the DEFSYS operand are allowed to IPL the NSS. This restriction allows the installation to ensure data integrity by restricting access to NSSs that might contain sensitive data:
  - NSSs that contain shared writable storage (defined with the SW and SN page descriptor codes)
  - CP NSS (defined with the SC page descriptor code), which may contain sensitive data
2. Virtual machines that share read/write pages, which are defined with the SW and SN page descriptor codes, are responsible for the integrity of the shared data. The operating system that uses these shared writable pages should ensure that these virtual machines are properly protected from one another.

#### Examples

```
CP DEFSYS CMS 0-D EW 20-23 EW F00-13FF SR MINSIZE=3M
MACHMODE XA,ESA,XC PARMREGS=0-15

defsys wss 0-8 ew 300-5ff sw 600-fff sn minsize=4m rstd

defsys newsys 0-4 ew e-21 ew 800-86f sr minsize=256k parmregs=0-15
machmode esa,xc
```

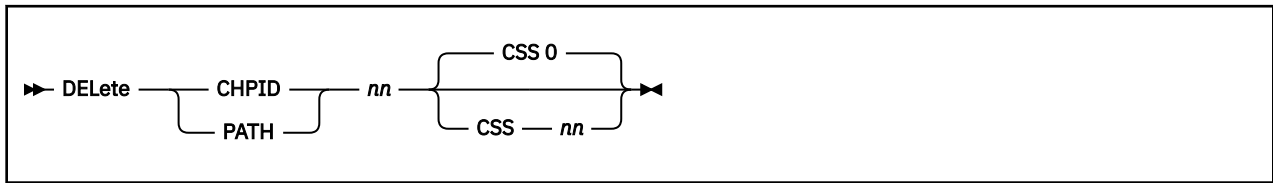
#### Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP263E Too many operands were supplied for this command.
- HCP299E The Named Saved System (NSS) *name* could not be defined because a class S file with the same name already exists.
- HCP339E The command processing cannot complete because the page descriptor code *invalid-code* for the page range *nnnnn-nnnnn* should be exclusive.

## DEFSYS

- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP427E *type* {*rdev/vdev/name[.userid]*} system spool space full; file {closed|held|purged}
- HCP435E DEFSYS command processing cannot complete because a paging I/O error occurred while writing a page.
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP440I Named Saved Segment (NSS) *name* was successfully defined in fileid *fileid*.
- HCP445E The Named Saved System (NSS) *name* could not be defined due to an unexpected error.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP847E Maximum system spool file limit exceeded
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*
- HCP1353E An invalid {name *name*|page range *nnnnn-nnnnn*} was detected in the DEFSYS command.
- HCP1354E An invalid storage page descriptor code *invalid-code* was detected for page range *nnnnn-nnnnn*.
- HCP1355E Segment number *nnn* was defined with both shared and exclusive page descriptor codes.
- HCP1356E *nnnnn-nnnnn* is a conflicting page range. Please check for overlaps and reissue the command.

## DELETE CHPID / PATH



### Authorization

Privilege Class: B

### Purpose

Use DELETE CHPID or PATH to delete an existing channel path identifier (CHPID) from the I/O configuration.

### Operands

**nn**

is the number of the channel path ID that you want to delete. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**CSS nn**

is the channel-subsystem ID from which the channel path is to be deleted. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'. See Usage Note “2” on page 259.

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require that the logical channel-subsystem facility be installed.

### Usage Notes

1. Before you can delete a channel path from your I/O configuration, you must:
  - a. Delete all of the devices attached to all the control units attached to that channel path using the DELETE DEVICE or IODEVICE command. (See “DELETE DEVICE / IODEVICE” on page 265.)
  - b. Delete all of the control units attached to that channel path using the DELETE CU or CNTLUNIT command (see “DELETE CU / CNTLUNIT” on page 263) or use the MODIFY CU or CNTLUNIT command (see “MODIFY CU / CNTLUNIT” on page 688) to delete channel path access.
  - c. Take that channel path logically offline using the VARY PATH command. (See “VARY PATH” on page 2053.)
  - d. Take that channel path physically offline using the VARY CHPID command. (See “VARY CHPID” on page 2046.)
2. On a machine with multiple logical channel-subsystem images, the DELETE CHPID and DELETE PATH commands are effective only for channel paths in a single channel-subsystem image. To delete channel paths that are in more than one channel-subsystem image, you must first enter MODIFY CHPID or MODIFY PATH commands to remove all but one channel-subsystem image. Then a DELETE CHPID or DELETE PATH command can be entered.
3. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.

## DELETE CHPID / PATH

- b. Run the IOCP utility against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
- c. Issue the SET IOCDS\_ACTIVE command to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR).

For more information, see [“IOCP” on page 2203](#) and [“SET IOCDS\\_ACTIVE” on page 1525](#).

4. For more information about deleting channel paths from your I/O configuration during initialization, see [Dynamically Deleting Channel Paths \(CHPIDs\) in z/VM: I/O Configuration](#).
5. To define a new channel path ID on the system, use the DEFINE CHPID or PATH command. See [“DEFINE CHPID / PATH” on page 132](#).
6. To change the definition of an existing reconfigurable channel path ID on the system, use the MODIFY CHPID or PATH command. (See [“MODIFY CHPID / PATH” on page 679](#).) There is no MODIFY command to change the definitions of the other types of channel paths. If you need to change their definitions, you must:
  - a. Delete the existing channel path definition, using the DELETE CHPID or PATH command.
  - b. Redefine the channel path definition, using the DEFINE CHPID or PATH command.
7. To display the status of a specific logical channel path and the devices with that logical channel path installed, or to display the type of a specific logical channel path, use the QUERY CHPID command. (See [“QUERY CHPID” on page 791](#).)
8. To display the physical status of all 256 of the machine's channel paths, use the QUERY CHPIDS command. (See [“QUERY CHPIDS” on page 796](#).)

**Note:** If the channel path is in single channel service (SCS) mode, it will appear as "not configured" in the display.
9. To display the logical and physical status of the channel paths installed on one or more devices, use the QUERY PATHS command. (See [“QUERY PATHS” on page 1064](#).)

## Responses

Response 1:

To delete channel path ID 32 from the I/O configuration, enter the following:

```
delete chpid 32
Channel path configuration change completed successfully
Ready;
```

Response 2:

To delete channel path ID 43 from the I/O configuration when it is currently defined only in channel-subsystem image 3, enter the following:

```
delete chpid 43 css 3
Channel path configuration change completed successfully
Ready;
```

## Messages

- HCP1006E Command processing cannot complete because the system does not support the logical-channel-subsystem facility.
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request [for channel path|for control unit|for device] failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem

- HCP6816E Dynamic I/O changes are not allowed {on this system|for CSS *nn*}
- HCP6818E Delete command failed. Channel path *nn* is physically available
- HCP6819E A dynamic I/O command is currently executing for user *userid*

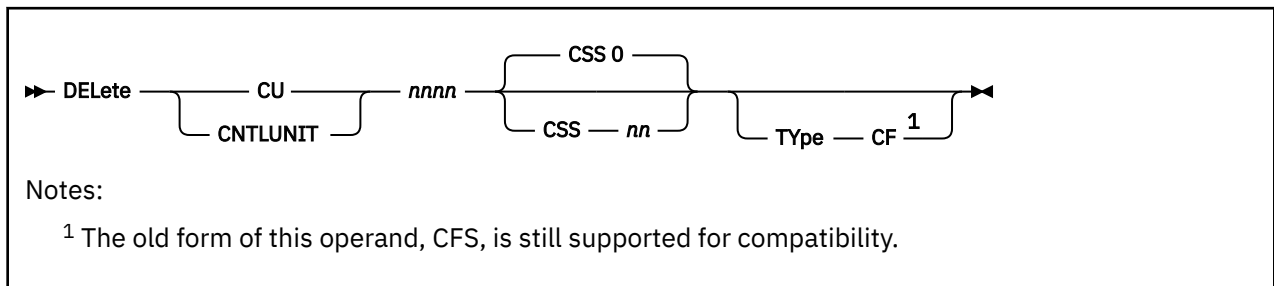
## DELETE CPUPOOL

---

### **Purpose**

DELETE CPUPOOL has been superseded by DELETE RESPOOL. See [“DELETE RESPOOL” on page 276](#).

## DELETE CU / CNTLUNIT



### Authorization

Privilege Class: B

### Purpose

Use DELETE CU or CNTLUNIT to delete an existing control unit from the I/O configuration.

### Operands

#### *nnnn*

is the number of the control unit that you want to delete. The variable *nnnn* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFE'.

#### **CSS** *nn*

is the channel-subsystem ID from which the control unit is to be deleted. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'. See Usage Note “2” on page 263.

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require that the logical channel-subsystem facility be installed.

#### **TYpe** **CF**

tells CP that this control unit is connected to coupling facility channel paths. Coupling facility channel paths include coupling facility sender (CFS), coupling facility receiver (CFR), coupling facility peer (CFP), cluster bus sender (CBS), cluster bus receiver (CBR), cluster bus peer (CBP), or internal coupling peer (ICP) channel paths.

### Usage Notes

1. Before you can delete a control unit, you must delete all devices attached to that control unit. To delete a device from a control unit, use the DELETE DEVICE or IODEVICE command. (See “DELETE DEVICE / IODEVICE” on page 265.)
2. On a machine with multiple logical channel-subsystem images, the DELETE CU and DELETE CNTLUNIT commands are effective only for control units in a single channel-subsystem image. To delete control units that are in more than one channel subsystem image, you must first enter MODIFY CU or MODIFY CNTLUNIT commands to remove all but one channel-subsystem image. Then a DELETE CU or DELETE CNTLUNIT command can be entered.
3. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.

## DELETE CU / CNTLUNIT

- b. Run the IOCP utility against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDs file to the SE hard disk. For more information, see [“IOCP” on page 2203](#).
  - c. Issue the SET IOCDs\_ACTIVE command to make the new IOCDs file the active file that the processor will use during the next power-on reset (POR). For more information, see [“SET IOCDs\\_ACTIVE” on page 1525](#).
4. For more information about deleting control units from your I/O configuration during initialization, see [z/VM: I/O Configuration](#).
  5. To define a new control unit on the system, use the DEFINE CU or CNTLUNIT command. (See [“DEFINE CU / CNTLUNIT” on page 161](#).)
  6. To change the definition of an existing control unit on the system, use the MODIFY CU command. (See [“MODIFY CU / CNTLUNIT” on page 688](#).)

## Responses

Response 1:

To delete control unit 300 from the I/O configuration, enter the following:

```
delete cu 300
Control unit configuration change completed successfully
Ready;
```

Response 2:

To delete control unit 43 from the I/O configuration when it is currently defined only in channel-subsystem image 3, enter the following:

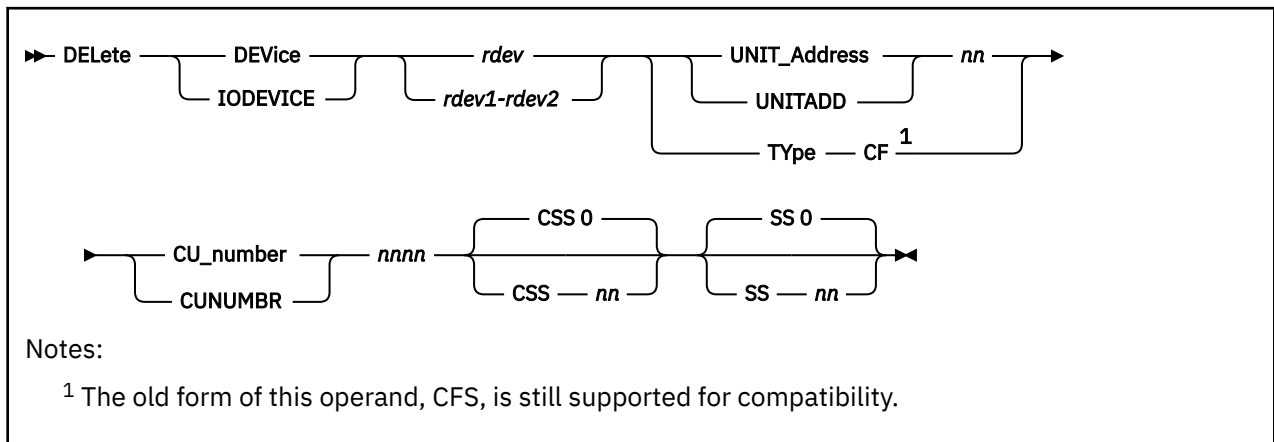
```
delete cu 43 css 3
Control unit configuration change completed successfully
Ready;
```

## Messages

- HCP1006E Command processing cannot complete because the system does not support the logical-channel-subsystem facility.
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request [for channel path|for control unit|for device] failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed {on this system|for CSS *nn*}
- HCP6819E A dynamic I/O command is currently executing for user *userid*



## DELETE DEVICE / IODEVICE



### Authorization

Privilege Class: B

### Purpose

Use DELETE DEVICE or IODEVICE to delete one or more existing real devices from the I/O configuration.

### Operands

**rdev**

**rdev1r-dev2**

is the real device number (or range of real device numbers) of the device that you want to delete. Each *rdev* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFF'.

**UNIT\_Address nn**

**UNITADD nn**

is the starting unit address associated with the device that you want to delete. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**TYpe CF**

tells CP that the specified devices are attached to a control unit that connects to coupling facility channel paths. Coupling facility channel paths include coupling facility sender (CFS), coupling facility receiver (CFR), coupling facility peer (CFP), cluster bus sender (CBS), cluster bus receiver (CBR), cluster bus peer (CBP), or internal coupling peer (ICP) channel paths.

**CU\_number nnnn**

**CUNUMBR nnnn**

is the number of one of the control units associated with the device that you are deleting. The variable *nnnn* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFE'.

**CSS nn**

is the channel-subsystem ID from which the device is to be deleted. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'.

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require that the logical channel-subsystem facility be installed.

**SS nn**

is the subchannel set with which this device is associated. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'03'.

**Note:** This parameter requires that the multiple-subchannel-set facility be installed.

## Usage Notes

1. Before you delete one or more real devices from your I/O configuration, they must all be offline and their corresponding subchannels must also be offline. To take a device offline, use the VARY (Real Device) command. To take a subchannel offline, use the VARY SUBCHANNEL command. For more information, see [“VARY \(Real Device\)” on page 2063](#) and [“VARY SUBCHANNEL” on page 2067](#).
2. On a machine with multiple logical channel-subsystem images, the DELETE DEVICE and DELETE IODEVICE commands delete the device with respect to the control unit's connection to channel paths within the single specified logical channel-subsystem image. To delete devices from more than one logical channel-subsystem image, you must enter a separate DELETE DEVICE or DELETE IODEVICE command for each logical channel-subsystem image from which you want the devices deleted.
3. When you delete one or more real devices from your I/O configuration, CP also deletes their corresponding real device blocks (RDEVs). If you want to delete the RDEV but do not want to change the I/O configuration, use the DELETE RDEVICE command. (See [“DELETE RDEVICE” on page 274](#).)
4. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.
  - b. Run the IOCP utility against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk. For more information, see [“IOCP” on page 2203](#).
  - c. Issue the SET IOCDS\_ACTIVE command to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR). For more information, see [“SET IOCDS\\_ACTIVE” on page 1525](#).
5. To define a new device on the system, use the DEFINE DEVICE or IODEVICE command. (See [“DEFINE DEVICE / IODEVICE” on page 167](#).)
  6. To change the definition of an existing device on the system, use the MODIFY DEVICE or IODEVICE command. (See [“MODIFY DEVICE / IODEVICE” on page 693](#).)
  7. To display the status of one or more real devices, use the QUERY (Real Device) command. (See [“QUERY \(Real Device\)” on page 1114](#).)

## Responses

Response 1:

To delete four real devices from your I/O configuration, enter the following:

```
delete device 300-303 unit_address 00 cu_number 300
Device configuration change completed successfully
RDEV 0300 deleted
RDEV 0301 deleted
RDEV 0302 deleted
RDEV 0303 deleted
Ready;
```

Response 2:

To delete four real devices in channel-subsystem image 3 and subchannel set 0 from your I/O configuration, enter the following:

```
delete device 440-443 unit_address 00 cu_number 400 css 3 ss 0
Device configuration change completed successfully
RDEV 0440 deleted
RDEV 0441 deleted
RDEV 0442 deleted
RDEV 0443 deleted
Ready;
```

## Messages

- HCP1006E Command processing cannot complete because the system does not support the logical-channel-subsystem facility.
- HCP1006E Command processing cannot complete because the system does not support the multiple-subchannel-set facility.
- HCP6805E Cannot determine partition being changed by this dynamic I/O request
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request for device failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed {on this system|for device *rdev*|for CSS *nn*}
- HCP6818E Delete command failed. Channel path *path* is physically available
- HCP6818E Delete command failed. Device *rdev* is a base Parallel Access Volume for which an alias exists
- HCP6818E Delete command failed. Device *rdev* is not subchannel offline
- HCP6819E A dynamic I/O command is currently executing for user *userid*

## DELETE EDEVICE



### Authorization

Privilege Class: B

### Purpose

Use DELETE EDEVICE to delete one or more emulated device definitions from the system.

### Operands

*rdev*

*rdev1-rdev2*

specifies the emulated devices that you want to delete. An emulated device in the active configuration can be entered as a 4-digit hexadecimal device number between X'0000' and X'FFFF'. For an emulated device in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF'. The leading digit specifies the subchannel set where the device is to be deleted. A range of devices cannot span subchannel sets.

### Usage Notes

1. Before you can delete an emulated device, you must vary offline the real device and subchannel that is associated with the emulated device. To VARY a device offline, use the VARY (Real Device) command. To VARY a subchannel offline, use the VARY SUBCHANNEL command.
2. To display the status of one or more real devices, use the QUERY (Real Device) command.
3. To display information about one or more emulated devices, use the QUERY EDEVICE command.
4. To define an emulated device, use the following commands:
  - a. SET EDEVICE command with the proper device characteristics, and the
  - b. VARY (Real Device) command to bring the device online.
5. You must delete all the aliases for an emulated device before you delete the base device.
6. You cannot delete the initial base for an emulated device until all aliases and any other bases are deleted.
7. The deletion of an NVME EDEVICE can be delayed for a device-specific interval to allow the NVME controller sufficient time to shut down.

### Responses

Response 1:

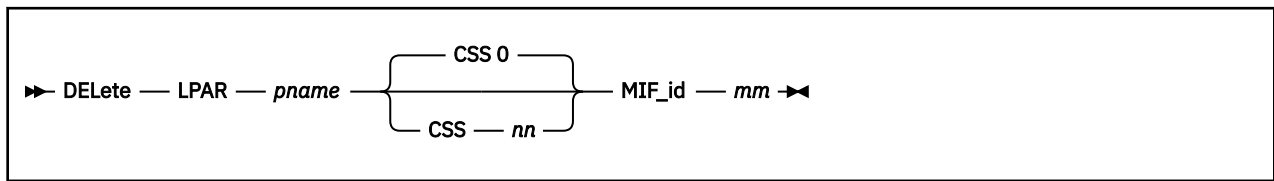
When you delete an emulated device 300 from your system, the command and response is as follows:

```
delete edevice 300
EDEV 300 was deleted.
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP009E Invalid range - *range*
- HCP040E Device *rdev* does not exist
- HCP6704E Missing token at end of line
- HCP6790I *rdev1* processing halted. The next device that would have been processed is *rdev2*.
- HCP6833E Device *rdev* is not subchannel offline
- HCP8702E Device *rdev* is defined as a real device.
- HCP8704E EDEV *rdev* was not deleted because the device could not be unloaded.
- HCP8704E EDEV *rdev* was not deleted because there are alias devices defined.
- HCP8704E EDEV *rdev* was not deleted because there are other base devices defined.

## DELETE LPAR



### Authorization

Privilege Class: B

### Purpose

Use the DELETE LPAR command to delete a logical partition from the I/O configuration.

### Operands

#### *pname*

specifies the name of the logical partition you are deleting. *pname* must be a 1-character to 8-character alphanumeric string.

**Note:** The first character of *pname* must be alphabetic and the following names cannot be used: PHYSICAL, REC, SYSTEM, PRIMnnn (where nnn are decimal digits).

#### *CSS nn*

is the channel-subsystem ID associated with the logical partition that is being deleted. The variable *nn* must be a 1-digit to 2-digit hexadecimal number between X'00' and X'0F.'

#### *MIF\_id mm*

is the MIF-ID within the specified CSS associated with the logical partition that is being deleted. The variable *mm* must be a 1-digit to 2-digit hexadecimal number between X'01' and X'0F.'

### Usage Notes

1. Once a logical partition is successfully deleted with this command, the MIF-ID within the CSS becomes an available slot in the configuration in which a logical partition can be subsequently defined.
2. This command requires that both the logical-channel-subsystem facility and the logical-partition-configuration facility be installed on your system.

### Examples

To delete logical partition VMSYSONE that is associated with CSS 1 and MIF-ID 3, enter the following:

```

delete lpar vmsysone css 1 mif 3
Logical partition configuration change completed successfully
Ready;
  
```

### Messages

- HCP864E Partition *pname* is not configured in CSS *nn*, MIF-ID *mm*
- HCP1006E Command processing cannot complete because the system does not support the logical-partition-configuration facility
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request for logical partition failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in channel subsystem

- HCP6816E Dynamic I/O changes are not allowed for CSS *nn*
- HCP6819E A dynamic I/O command is currently executing for user *userid*

## DELETE PCIFUNCTION

```
▶▶ DELEte — PCIFunction — rpfid ▶▶
```

### Authorization

Privilege Class: B

### Purpose

Use DELETE PCIFUNCTION to delete a real PCI function from the I/O configuration.

### Operands

#### *rpfid*

is the real function identifier of the PCI function you are deleting from the I/O configuration. The *rpfid* must be a 1- to 8-digit hexadecimal number between X'00000000' and X'7FFFFFFF'. The maximum value of the *rpfid* is model dependent and your machine likely supports a lower maximum value.

### Usage Notes

1. Before you delete a PCI function from your I/O configuration, and the PCIF is configured to the z/VM system you are issuing the DELETE command from, it must be offline before the DELETE PCIF is issued. To take a PCI function offline, use the VARY OFF PCIFUNCTION command. If the PCI function is attached to a guest, then the DETACH PCIFUNCTION command must be issued before the VARY OFF PCIFUNCTION command.
2. If a PCIF is to be deleted from the I/O configuration and the PCIF is configured to a different z/VM system in the I/O configuration than the z/VM system where the command is issued, the PCIF will be forcefully detached from any users and forcefully varied offline. So care must be used when using the DELETE PCIF command.
3. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.
  - b. Run the IOCP utility against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
  - c. Issue the SET IOCDS\_ACTIVE command to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR).
4. For more information about deleting PCI functions from your I/O configuration, see [Adding, Deleting, or Changing a PCI Function in z/VM: I/O Configuration](#).
  5. To define a PCI function to the system, use the DEFINE PCIFUNCTION command.
  6. To change the definition of an existing PCI function on the system, use the MODIFY PCIFUNCTION command.
  7. To display the status of one or more real PCI functions, use the QUERY PCIFUNCTION command.

### Responses



Response 1:

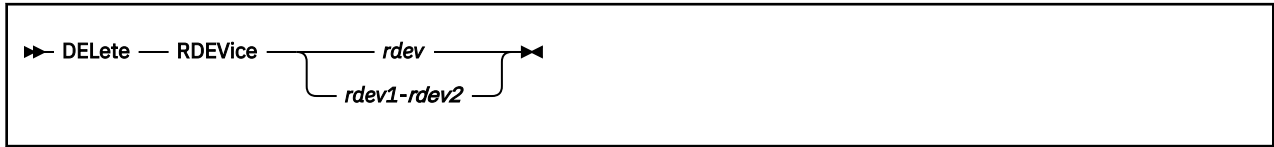
To delete a PCI function from your configuration, enter the following:

```
del pcif 43
PCI function configuration change completed successfully
Ready;
```

## Messages

- HCP1006E Command processing cannot complete because the system does not support the Dynamic-I/O-for-I/O-adapters facility.
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request for PCI function failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6818E Delete command failed. PCI function *rpfid* is not offline.
- HCP6819E A dynamic I/O command is currently executing for user *userid*

## DELETE RDEVICE



### Authorization

Privilege Class: B

### Purpose

Use DELETE RDEVICE to delete one or more real device control blocks (RDEVs) on the system. An RDEV is a piece of storage associated with a real device that contains information about that device's features and status. Thus, when you issue the DELETE RDEVICE command, you are actually deleting a piece of storage, not an actual real device.

To delete an actual real device, use the DELETE DEVICE or IODEVICE command (see [“DELETE DEVICE / IODEVICE”](#) on page 265).

### Operands

*rdev*

*rdev1-rdev2*

is the number of the real device whose RDEV you want to delete. A real DASD in the active configuration can be entered as a 4-digit hexadecimal device number between X'0000' and X'FFFF'. For a real DASD in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is to be deleted. A range of devices cannot span subchannel sets.

### Usage Notes

1. Before you can delete a real device control block (RDEV), you must take the device and subchannel associated with this RDEV offline. To take a device offline, use the VARY (Real Device) command (see [“VARY \(Real Device\)”](#) on page 2063). To take a subchannel offline, use the VARY SUBCHANNEL command (see [“VARY SUBCHANNEL”](#) on page 2067).
2. To display the status of one or more real devices, use the QUERY (Real Device) command (see [“QUERY \(Real Device\)”](#) on page 1114).

**Note:** The QUERY (Real Device) command displays the status of the device only if you issue the QUERY command in the logical partition that owns the device. If you issue the QUERY command in one of the other logical partitions, CP gives you a message stating that the device does not exist.

3. To create a real device control block, use the:
  - a. SET RDEVICE command (see [“SET RDEVICE”](#) on page 1627) with the proper device characteristics, and
  - b. VARY (Real Device) command (see [“VARY \(Real Device\)”](#) on page 2063) to bring the device online.

### Responses

Response 1:

To delete real device control block 300 from your system, enter the following:

```
delete rdevice 300  
RDEV 300 deleted  
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP009E Invalid range - *range*
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP040E Device *rdev* does not exist
- HCP1150E DASD *rdev* is not a valid base exposure.
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*
- HCP6790I *rdev1* processing halted. The next device that would have been processed is *rdev2*.
- HCP6833E Device *rdev* is not subchannel offline
- HCP6862E Device *rdev* cannot be changed or deleted because it is a base Parallel Access Volume for which an alias exists.

## DELETE RESPOOL

```
►► DELeTe — RESPool — poolname ►◄  
      └── CPUPOOL ─┘
```

### Authorization

Privilege Class: A

### Purpose

Use DELETE RESPOOL to delete a resource pool that is already defined. For compatibility with prior releases, CPUPOOL is accepted as a synonym for RESPOOL.

### Operands

#### *poolname*

is the name of the resource pool to be deleted.

### Usage Notes

1. The DELETE RESPOOL command can be issued only for a resource pool that has already been created with the DEFINE RESPOOL command.
2. The resource pool can be deleted only if there are no users assigned to the pool. Use the SCHEDULE command to remove users from the resource pool.

### Responses

Response 1:

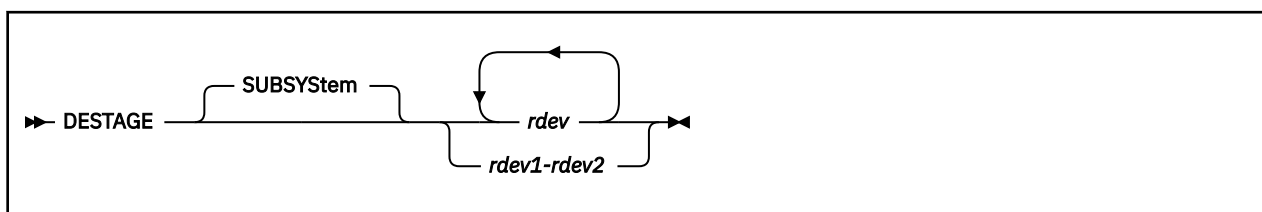
```
Resource pool poolname is deleted
```

confirms that the resource pool has been deleted.

### Messages

- HCP002E Invalid operand — *operand*
- HCP003E Invalid option — command contains extra option(s) starting with *option*
- HCP1001E An operand is missing for RESPOOL
- HCP3151E Resource pool *poolname* does not exist
- HCP3152E Resource pool *poolname* cannot be deleted because users are still assigned to it

## DESTAGE



### Authorization

Privilege Class: B

### Purpose

Use DESTAGE to force the transfer of modified data in a direct access subsystem cache or nonvolatile storage (NVS) to the backing storage device. This command is valid for cache storage subsystems that support the cache fast write or DASD fast write functions.

The DESTAGE command can be used as part of a shutdown process.

### Operands

#### SUBSYSstem

specifies that data for the entire storage subsystem is to be destaged, regardless of which device within the subsystem receives the destage order. This is the default (and only) value.

#### rdev

#### rdev1-rdev2

is the real device number, a list of real device numbers, or a range of real device numbers, of devices to which the destage order is to be sent.

### Usage Notes

1. This command may produce delayed responses. For more information, see [“Delayed Responses to CP Commands”](#) on page 7.
2. The DESTAGE command is only valid for cache storage subsystems that support cache fastwrite or DASD fastwrite functions. An error message is generated if the subsystem containing the specified device does not support any of these functions.

### Responses

Response 1:

*Immediate Response:* If the transfer of subsystem data occurs immediately, the following response is issued to the originator of the command. This response is repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev Subsystem data has been destaged.
```

#### rdev

is the real device number of one of the specified devices in a subsystem being destaged.

Response 2:

*Delayed Response:* If the transfer of subsystem data does not occur immediately, the following response is issued:

```
Command started: DESTAGE rdev.
```

## DESTAGE

When the transfer of subsystem data completes, the following responses are issued:

```
rdev Subsystem data has been destaged.
```

Under certain circumstances, it is possible that a cache storage subsystem will not notify CP when it has completed the transfer of the subsystem data. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing is ended after generating the following response:

```
Command results lost: DESTAGE rdev.
```

If errors occur during the asynchronous processing of the command that prevent the successful completion of the command, the following response is displayed:

```
Command failed: DESTAGE rdev.
```

### ***rdev***

is the real device number of one of the specified devices in a subsystem being destaged.

Response 3:

*Response to System Operator:* If the originator of the command is not the system operator, the following response is issued to the system operator when the transfer of subsystem data completes. This response is also repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev Subsystem data was destaged by userid.
```

### ***rdev***

is the real device number of one of the specified devices in a subsystem being destaged.

### ***userid***

is the ID of the user who issued the command.

## Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP009E Invalid range - *range*
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*
- HCP1109E The *command* command cannot be processed for device *rdev* because the device is the secondary in a duplex pair.

## DETACH

---

### Purpose

Use the DETACH commands to detach the following:

- Virtual processors from your virtual machine
- Real devices from the host system
- Logical devices from the host system
- Real devices from your virtual machine
- Logical devices from your virtual machine
- Virtual devices from your virtual machine
- Virtual message processor and message devices from your virtual machine
- A guest LAN from the host system.

### Usage Notes

(General for DETACH)

1. SET IMSG OFF will suppress the informational messages for DETACH CPU, DETACH MSGPROC, and DETACH VDEV. DETACH RDEV is not affected by the IMSG setting.
2. You cannot be certain that the device has been detached until you receive the appropriate response from the system. If after a reasonable amount of time you haven't received the expected response, enter the QUERY command to determine the status of the device. Detachment may be pending or may have been unsuccessful.
3. If you detach a minidisk with this command without releasing it, CMS implicitly releases the minidisk. QUERY DISK will not display that minidisk.
4. You can designate a virtual CTCA by device address. If a virtual CTCA is detached while coupled with another virtual CTCA, the remote device is left in an uncoupled state, similar to its original condition after the DEFINE command.
5. In an SSI cluster, MDC will be turned on and off automatically when RDEV SHARED NO is in effect.
6. The DETACH command will not affect the various draining status indicators, so you can set the desired draining status before you attach a DASD to the system.

### Messages

(General for DETACH)

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*
- HCP006E Invalid device type *vdev*
- HCP007E Invalid userid - *userid*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP045E *userid* not logged on

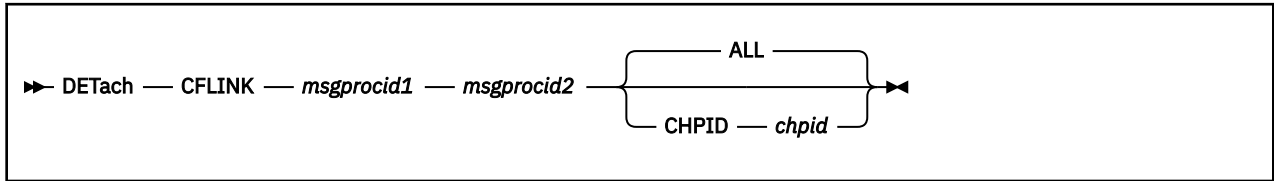
## DETACH

- HCP121E *type {rdev|ldev}* not attached to *{userid|SYSTEM}*
- HCP121E SYSASCII not attached to you
- HCP121E SYSASCII not attached to *userid*
- HCP124E DASD *rdev* in use by *nnnnn* users
- HCP143E *type rdev* in use by system
- HCP260E DETACH command processing cannot complete due to errors
- HCP263E Too many operands were supplied for this command.
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP404E Device *rdev* not detached - device is being given
- HCP427E *type {rdev|vdev|name [userid]}* system spool space full; file {closed|held|purged}
- HCP429E *type vdev* spool error; file {held|purged}
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP475I Fatal I/O error trying to read directory from *valid* { for user *userid*}
- HCP1100E A logical device number was not supplied or it is invalid.
- HCP1119E Device *rdev* was not detached. Detach with the ALL option is only valid for devices dedicated to a user or for tapes attached MULTIUSER.
- HCP1120E The requested DETACH for device *{vdev|ldev}* did not complete in the allotted time.
- HCP1121E The DETACH command has been canceled for {device *{rdev|ldev}* | devices *{rdev-rdev|ldev-ldev}*}.
- HCP1150E DASD *{rdev|vdev}* is not a valid base exposure.
- HCP1252I Device *rdev* has been unassigned by user *userid*
- HCP1280E Device *vdev* not usable; reserve lost [by *userid* as *vdev*]
- HCP1281I Path *channel* to device *device* now offline; path group ID mismatch.
- HCP1401I Expanded storage is not supported.
- HCP1454E CPU *nn* does not exist.
- HCP1456E The base CPU *nn* cannot be detached.
- HCP1809E Device *vdev* cannot be detached from the user because it is a VDISK and Live Guest Relocation is active.
- HCP2783E LAN *ownerid lanname* does not exist
- HCP2793E Device *vdev* not detached; DETACH NIC *vdev* to remove network devices *vdev-vdevlast*
- HCP2793E Device *vdev* not detached; SET VSWITCH *switchnm* DISCONNECT to remove network devices *base-last*
- HCP2793E Device *vdev* not detached; SET VSWITCH *switchnm* BRIDGEPORT DISCONNECT to remove network devices *base-last*
- HCP2795E You are not authorized to DETACH LAN *lanname* for OWNERID *ownerid*
- HCP2805E Message processor *msgprocid* does not exist in your I/O configuration.
- HCP2807E Must detach message processor *msgprocid* to remove message device *nnnn*.
- HCP3049E An internal error occurred; the state of the virtual switch *switchname* may be unstable.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6005E Option *option* is not supported by z/VM.
- HCP6324I An unexpected condition occurred writing to the subchannel for device *rdev*, code = *code*.
- HCP6866E Device *rdev* cannot be detached from the system because it is a base Parallel Access Volume for which one or more alias volumes are attached to the system.
- HCP6866E Device *vdev* cannot be detached from the user because it is a base Parallel Access Volume for which one or more alias volumes are associated.



- HCP6866E Device *rdev* cannot be detached from the user because it is a base Parallel Access Volume for which one or more alias volumes are attached.

## DETACH CFLINK



### Authorization

Privilege Class: B, G

### Purpose

Use the DETACH CFLINK command to detach links between coupling facilities.

### Operands

#### *msgprocid1*

is the user ID of the sending coupling facility virtual machine.

#### *msgprocid2*

is the user ID of the receiving coupling facility virtual machine.

#### **ALL**

specifies that all links, where *msgprocid1* is the sender and *msgprocid2* is the receiver, should be deleted.

#### **CHPID** *chpid*

specifies that only the link from *msgprocid1* to *msgprocid2* that is known to *msgprocid2* (the receiver) through channel path ID *chpid* should be deleted.

### Usage Notes

1. A class G only user must be coupled to both the sending and receiving coupling facilities.
2. If you detach a coupling facility link, all message devices that were created in the sender when that link was created are also deleted.

### Examples

CFVM1 and CFVM2 are connected as follows:

- In CFVM1, channel path ID E0 (devices 400 and 401) leads to channel path ID F0 in CFVM2.
- In CFVM1, channel path ID A0 (devices 800 and 801) leads to channel path ID B0 in CFVM2.

To detach all links, enter the following:

```
detach cflink cfvm1 cfvm2
```

To detach just the first link, enter the following:

```
detach cflink cfvm1 cfvm2 chpid f0
```

(This also results in devices 400 and 401 being deleted in CFVM1.)

To detach just the second link, enter the following:

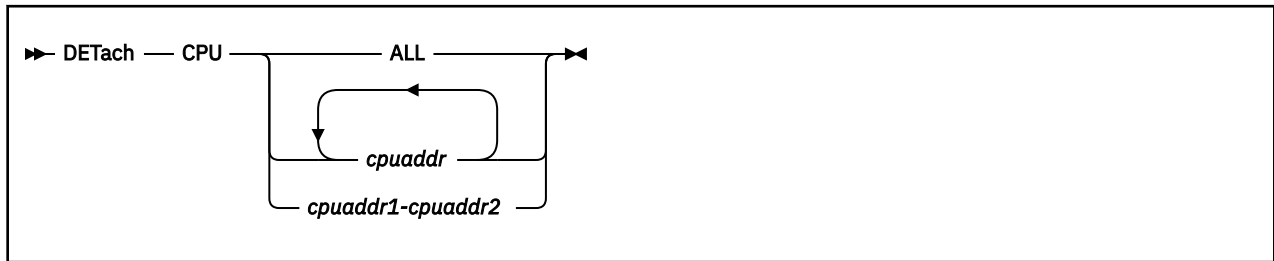
```
detach cflink cfvm1 cfvm2 chpid b0
```

(This also results in devices 800 and 801 being deleted in CFVM1.)

## Messages

- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP047E CFLINK with this CHPID does not exist
- HCP047E CFLINK in *coupling\_facility\_id* does not exist
- HCP260E Command processing cannot complete due to errors.
- HCP846E Invalid channel path identifier.
- HCP2800E *msgprocid* is not a CF Service Machine.
- HCP2805E One or more message processors does not exist in your I/O configuration.
- HCP2806E You are not allowed to control message processors.

## DETACH CPU



### Authorization

Privilege Class: G

### Purpose

Use DETACH CPU to remove processors from your virtual machine configuration.

### Operands

#### ALL

specifies all virtual processors currently defined.

#### *cpuaddr*

is the processor address of the processor that you want to detach from your virtual machine configuration.

#### *cpuaddr cpuaddr*

is a list of addresses and/or ranges delimited by spaces.

#### *cpuaddr1-cpuaddr2*

specifies a nonwrapping pair of addresses separated by a hyphen (-) and not separated by any blanks.

### Usage Notes

1. You can detach any virtual processor from your virtual machine configuration except the base processor. You cannot detach the base processor from your virtual machine system.
2. Whenever you detach a virtual processor, your virtual machine is reset as if a CP SYSTEM CLEAR command had been processed. The effects of this system-reset-clear function are described in detail in the description for the SYSTEM CLEAR command. The system-reset-clear function sets all of your virtual machine's storage to binary zeros. Therefore, you must repeat your IPL procedure to proceed.
3. SET IMSG OFF suppresses informational messages for this command.
4. For more information, see [“Usage Notes” on page 279](#).

### Responses

```

CPU nn DETACHED
  
```

indicates that the command has been successfully processed.

This response is followed by:

```

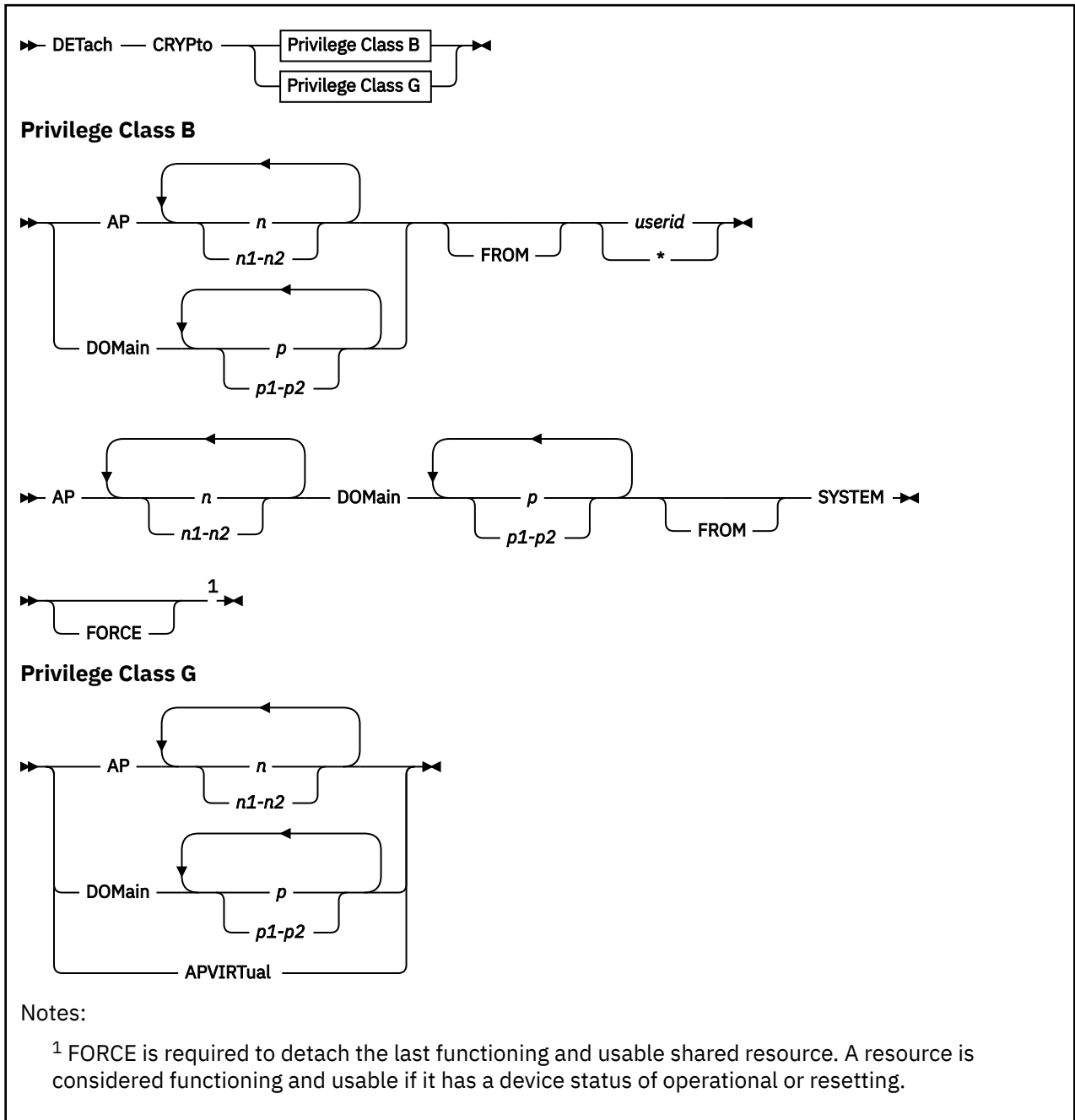
STORAGE CLEARED - SYSTEM RESET
  
```

indicating that all of your virtual machine's storage has been cleared to binary zeros, and your processors and I/O configuration have been reset.

***nn***

identifies the number of the CPU currently defined.

## DETACH CRYPTO



### Authorization

Privilege Class: B, G

### Purpose

Use DETACH CRYPTO to:

- Remove dedicated crypto resources from a virtual machine.
- Remove shared crypto resources from the system.
- Remove a virtual machine's access to the system's shared crypto resources.

## Operands

***n***

***n1-n2***

specifies the adapter number, a list of adapter numbers, a range of adapter numbers or a combination of these, to be detached. When detaching from a userid, crypto resources on all domains on the specified adapters will be detached. If you are specifying a list of adapter numbers, insert at least one blank between the numbers you specify. If you are specifying a range of adapter numbers, insert a hyphen (-) between the numbers you specify. The adapter number is a decimal number between 0 and 255.

***p***

***p1-p2***

specifies the domain number, a list of domain numbers, a range of domain numbers or a combination of these, to be detached. When detaching from a userid, crypto resources on all adapters on the specified domain(s) will be detached. If you are specifying a list of domain numbers, insert at least one blank between the numbers you specify. If you are specifying a range of domain numbers, insert a hyphen (-) between the numbers you specify. The crypto domain number is a decimal number between 0 and 255.

***userid***

**\***

is the virtual machine that the crypto resources are being detached from. When you are detaching from your own virtual machine, you can specify your user ID as an asterisk (\*).

### SYSTEM

indicates that the crypto resources are to be detached from the system.

### FORCE

this option is required when detaching the last remaining operational shared crypto resource from the system.

### APVIRTual

removes a virtual machine's access to the system's shared crypto resources.

## Usage Notes

1. When detaching dedicated crypto resources from a virtual machine configuration, the DETACH command must specify only adapter(s), or only domain(s). The DETACH command will be rejected if both adapters and domains are specified.
2. The crypto resources being detached are processed individually. A message indicating success or failure is given for each resource processed by the DETACH CRYPTO command. If the FORCE option is not specified, all of the resources specified which do not require the FORCE option will be detached. The detach will fail for any resource that requires the FORCE option.
3. For information on planning and managing crypto resources on a z/VM system, see [Crypto Planning and Management in z/VM: CP Planning and Administration](#).

## Examples

To detach adapter 7 domains 3 and 4 from the shared pool, enter:

```
DETACH CRYPTO AP 7 DOMAIN 3-4 FROM SYSTEM
```

To detach all dedicated domains on adapter 1 from the virtual machine, enter:

```
DETACH CRYPTO AP 1 FROM userid
```

To detach your access to the shared crypto resources, enter:

```
DETACH CRYPTO APVIRTUAL
```

## Responses

Response 1:

This response is sent to the command issuer when the command issuer is Class G.

```
Crypto AP apnum Domain domnum detached
Crypto APVIRTUAL detached
```

Response 2:

This response is sent to the command issuer when the command issuer is Class B.

```
Crypto AP apnum Domain domnum detached from target
Crypto AP apnum Domain domnum detached from SYSTEM
Crypto AP apnum Domain domnum detached from SYSTEM with FORCE
```

Response 3:

This response is sent to the primary system operator when the primary system operator is not the command issuer.

```
Crypto AP apnum Domain domnum detached from target by originator
Crypto AP apnum Domain domnum detached from SYSTEM by originator
Crypto AP apnum Domain domnum detached from SYSTEM with FORCE by originator
```

Response 4:

This response is sent to the target when the target is not the command issuer or the primary system operator.

```
Crypto AP apnum Domain domnum detached by originator
```

### ***apnum***

is the adapter number of the crypto resource.

### ***domnum***

is the domain number of the crypto resource.

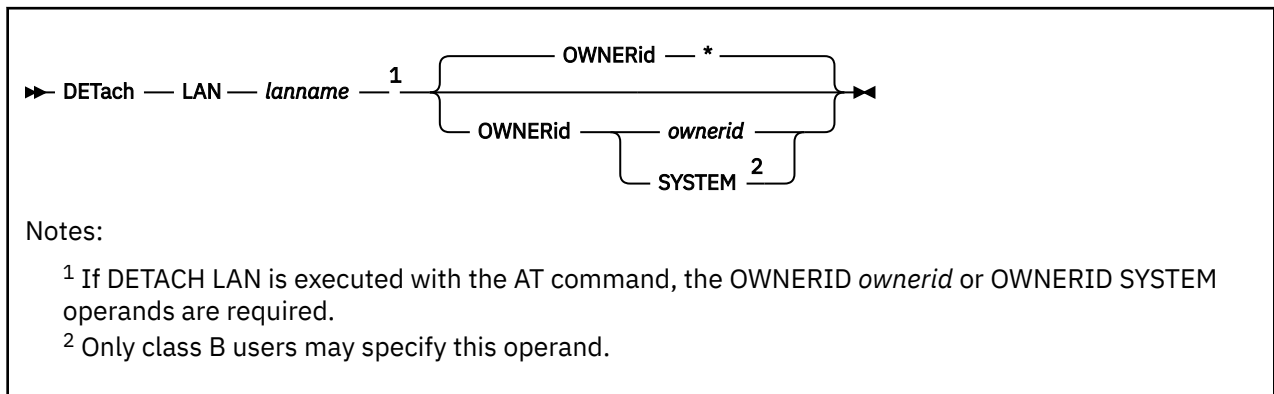
## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*| command contains extra option(s) starting with *option*
- HCP045E *userid* not logged on
- HCP1014E A required option is missing
- HCP1726E User *userid* has no dedicated AP crypto resources to detach.
- HCP1726E User *userid* has no dedicated AP crypto resources to detach on AP *apnum*.
- HCP1726E User *userid* has no dedicated AP crypto resources to detach on domain *domnum*.
- HCP1727E Crypto AP *apnum* Domain *domnum* is not attached to SYSTEM.
- HCP1727E There are no shared AP crypto resources to detach from SYSTEM.
- HCP1728E AP Crypto resources are not installed on this system.
- HCP1728E Crypto AP *apnum* Domain *domnum* is not assigned to this system.
- HCP1740E Crypto APVIRTUAL is not defined.
- HCP1744E The FORCE option is required to detach AP *apnum* Domain *domnum* because it is the last shared AP crypto resource.
- HCP1744E The FORCE option is required to detach AP *apnum* Domain *domnum* because it is the last usable crypto resource in the shared pool.
- HCP1831E The command could not be completed because of a temporary condition. Try again later.
- HCP2768E Missing user ID



- HCP2768E Missing crypto domain number
- HCP2768E Missing crypto AP number
- HCP6706E Invalid user ID
- HCP6706E Invalid Crypto Domain number
- HCP6706E Invalid Crypto AP number

## DETACH LAN



### Authorization

Privilege Class: B, G

### Purpose

Use DETACH LAN to eliminate a guest LAN from the CP system. This operation removes the designated LAN from the System LAN Table, disconnects any virtual adapters using the LAN, and releases system resources associated with the LAN. In general, this command must be issued by the owner of the LAN.

The Class B form of the command allows the invoker to remove a LAN owned by another user. Only a Class B user can specify OWNERID SYSTEM (removing a SYSTEM LAN).

### Operands

#### *lanname*

is the name of the guest LAN to be eliminated. Both *lanname* and *ownerid* must be known to fully qualify a LAN.

#### OWNERid \*

#### OWNERid *ownerid*

#### OWNERid SYSTEM

specifies the owner of the LAN. When OWNERID \* is specified, it indicates the invoker is the owner of the LAN. This is the default when OWNERID is omitted.

The Class G form of the command only accepts \* or the user ID of the invoker.

The Class B form of the command accepts another user ID (like OWNERID SYSTEM).

### Usage Notes

1. Only the current owner of the LAN, or a Class B user, can DETACH the LAN.
2. Any virtual network adapter that is connected to the LAN when it is detached is automatically uncoupled.

### Examples

1. DETACH LAN - Class G:

To remove the user LAN named QDIO0 from the system, enter:

```
cp detach lan qdio0
LAN VMUSERX QDIO0 is destroyed
```

2. DETACH LAN - Class B:

To remove the SYSTEM LAN named INEWS from the system, enter:

```
cp detach lan inews ownerid system  
LAN SYSTEM INEWS is destroyed
```

## Responses

Response 1:

```
LAN ownerid lanname is destroyed
```

is the normal response for this command.

Response 2:

```
NIC vdev is disconnected from LAN ownerid lanname
```

is the response for this command when a virtual network adapter connected to a LAN is detached.

## DETACH MSGPROC

►► DETach — MSGProc — *msgprocid* ◄◄

### Authorization

Privilege Class: G

### Purpose

Use the DETACH MSGPROC command to detach the specified message processor and message devices from the issuer's I/O configuration.

### Operands

#### *msgprocid*

specifies the user ID of the CF Service Virtual Machine to be deleted. CP will break the Coupling Facility connections between the user ID that issues this command and the user ID specified by *msgprocid*.

### Usage Notes

1. To define a virtual message processor and associated virtual message devices, use the DEFINE MSGPROC command.
2. To display information about virtual message processors in your I/O configuration, use the QUERY VIRTUAL MSGPROC command.
3. To display information about the virtual message devices within your I/O configuration, use the QUERY VIRTUAL MSGDEVICES command.
4. To restart a virtual message processor that is within your I/O configuration, use the RESTART MSGPROC command.
5. To remove all message facility capability from your virtual machine, use the SET MSGFACIL OFF command.
6. SET IMSG OFF suppresses informational messages for this command.
7. For more information, see DETACH (general) Usage Notes.

### Responses

Response 1:

To detach the CFCC1 message processor and associated message devices from your I/O configuration, enter the following:

```
detach msgp cfcc1
MSGPROC CFCC1 DETACHED
Ready;
```

## DETACH NIC

---

```
▶▶ DETach — NIC — vdev ▶◀
```

### Authorization

Privilege Class: G

### Purpose

Use DETACH NIC to remove a virtual network adapter (network interface card) from your virtual machine configuration. This command disconnects the virtual adapter from the guest LAN (if necessary), and removes each virtual device that was created by the DEFINE NIC command for this adapter.

### Operands

#### *vdev*

specifies the virtual device number (hex) of the adapter to be detached. You may specify the virtual device number of any virtual device that is attached to the adapter. However, CP responses will refer to each virtual adapter by its *base* device address; this is the first device in the range.

### Responses

Response 1:

```
NIC vdev is destroyed; devices vdev-lastvdev detached
```

#### *vdev*

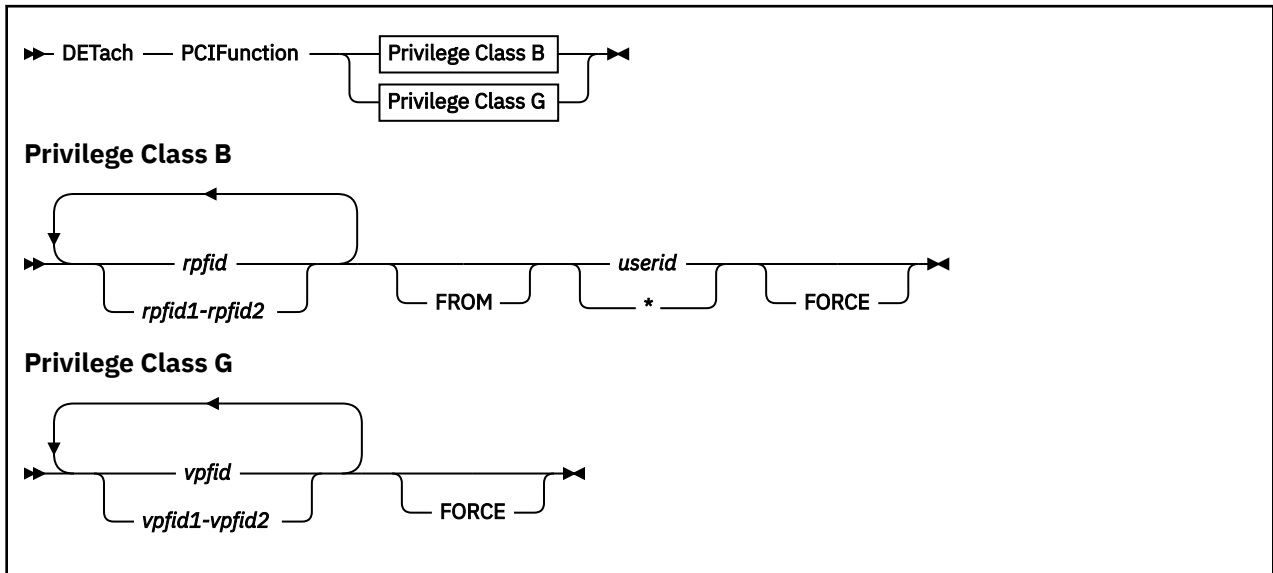
specifies the base device address

#### *lastvdev*

specifies the last device that is defined as part of this adapter

is the normal response for this command.

## DETACH PCIFUNCTION



### Purpose

Use DETACH to detach PCI function(s) from a virtual machine. Ordinarily, the guest is signaled to quiesce its use of the function. The function is detached and the command completes only when the guest relinquishes the function or a grace period expires.

### Operands

#### *rpfid*

#### *rpfid1-rpfid2*

specifies a real PCI function ID or a range of PCI function IDs to be detached from the specified user. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. The total number of PCI functions specified cannot exceed 256.

#### *vpfid*

#### *vpfid1-vpfid2*

specifies a virtual PCI function ID or a range of PCI function IDs to be detached from the specified user. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. The total number of PCI functions specified cannot exceed 256.

#### **FROM** *userid*

identifies the virtual machine to which the function or functions are currently attached. The specified user must currently be logged onto the system and not in the process of logging on. If you are detaching a function from your own virtual machine, you may specify your userid as an asterisk (\*). If you are a class G user you may only detach PCI functions from your own userid.

#### **FORCE**

indicates the function will be detached from the guest with no signal to the guest that the function is being detached and no time given to the guest to gracefully complete use of the function.

### Usage Notes

1. If the FORCE operand is not specified, the guest must be running in order to gracefully complete use of the function. For example, if the guest is stopped with CP SET RUN OFF, the guest will not be able to complete its use of the function.

## Responses

The receiver of the PCI function detached, the user who issued the DETACH command, and the primary system operator all receive messages indicating the DETACH was successful.

Response 1:

The following response goes to the originator of the DETACH command:

```
PCI FUNCTION rpfid DETACHED userid vpfid
```

Response 2:

The following response goes to the detachee of the PCI function:

```
PCI FUNCTION vpfid DETACHED BY originator
```

Response 3:

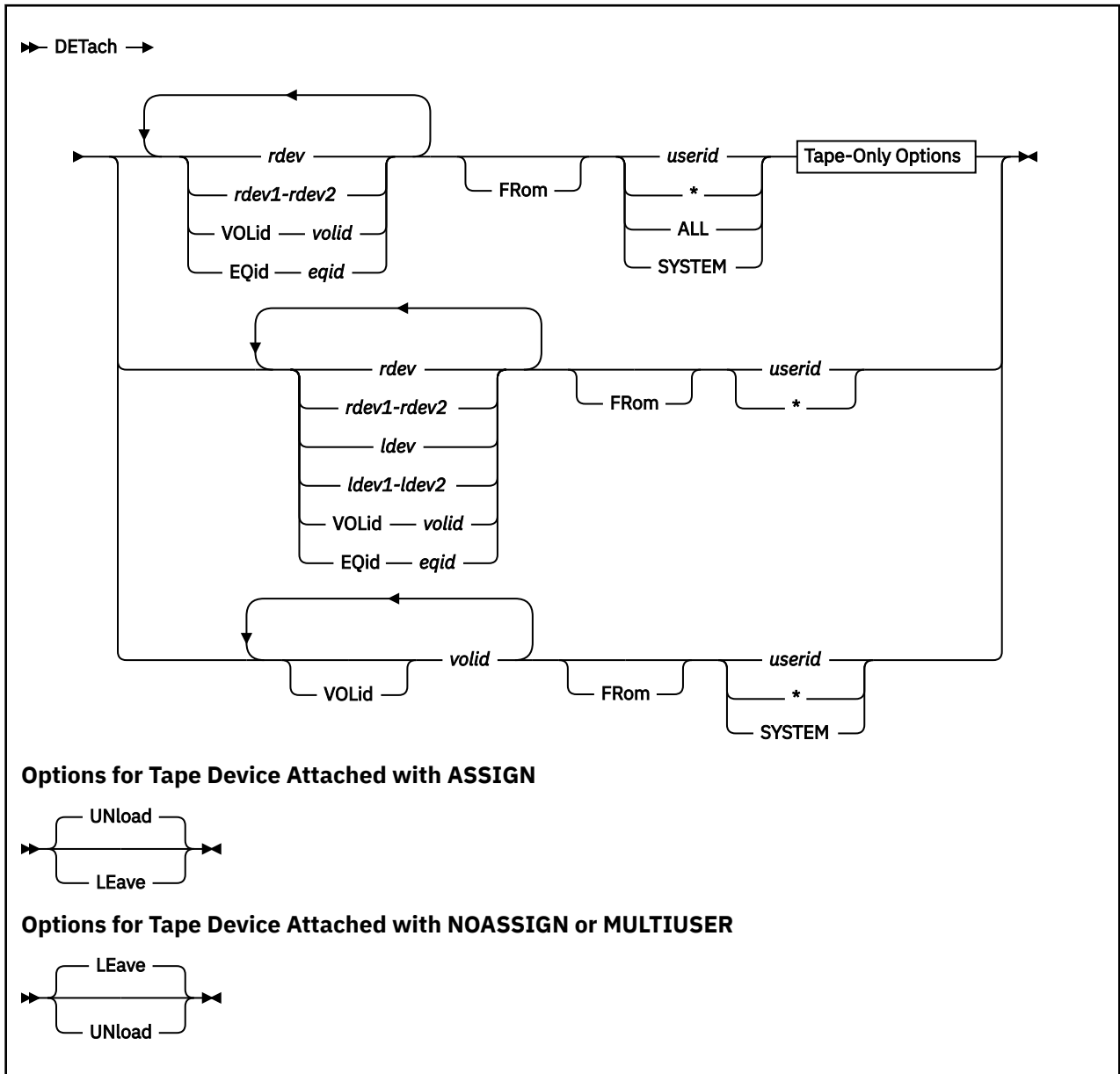
The following responses goes to the system operator:

```
PCI FUNCTION rpfid DETACHED FROM userid vpfid BY originator
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP021E A real PCI function ID was not supplied or it is invalid.
- HCP022E A virtual PCI function ID was not supplied or it was invalid.
- HCP040E PCI function *vpfid* does not exist
- HCP040E PCI function *rpfid* does not exist
- HCP121E PCI function *rpfid* not attached to *userid*
- HCP6000E The range of PCI function IDs cannot exceed 256.

## DETACH (Real Device)



### Authorization

Privilege Class: B

### Purpose

Use DETACH (real device) to detach the following:

- Local or remote real devices from one or more virtual machines
- Local real devices from the host system

**Note:** When detaching a remote device, specify its logical device address.



## Operands

### *rdev*

#### *rdev1-rdef2*

is a real device number, a list of real device numbers, or a range of real device numbers to be detached from the indicated user, from all users, or from the system. On a single command line, you may combine device numbers, lists of device numbers, and ranges of device numbers. Specify a real device number as one to four hexadecimal digits.

### *ldev*

#### *ldev1-ldev2*

is a logical device number, a list of logical device numbers, or a range of logical device numbers to be detached from the indicated user. On a single command line, you may combine device numbers, lists of device numbers, and ranges of device numbers. Specify a logical device number as an L followed by one to four hexadecimal digits.

### **VOLid**

indicates that a volume serial number, which must follow this keyword, is used to identify the real device. If the volume serial number is valid as an operand of the command (e.g., is a valid device number), specify VOLid to indicate that the operand is a volume serial.

### *valid*

specifies the 1- through 6-character volume serial number of a real DASD.

### **EQid**

indicates that a device equivalency ID, which must follow this keyword, is used to identify the real device. If the EQID is valid, and only one device with the specified EQID is attached to the target, the detach will be performed. If more than one device with the specified EQID is attached to the target, no detach will be performed.

### *eqid*

specifies a string of 1–8 alphanumeric characters for a user-specified EQID or a string of 50 alphanumeric characters plus a dash ("-") for a system-generated EQID.

### *userid*

\*

identifies the virtual machine to which the device is currently attached. The specified user must be logged on, not in the process of logging on. If you are detaching a device from your own virtual machine, you may specify *userid* as an asterisk (\*).

### **ALL**

detaches a real device from all virtual machines without having to specify the user IDs. This operand can be used to detach devices attached with the MULTIUSER option. The ALL operand is not valid for devices attached to SYSTEM.

### **SYSTEM**

detaches a DASD currently being used by CP for allocation and control.

### **UNload**

detaches the tape device and automatically rewinds and unloads it. UNLOAD is the default if the tape was attached with the ASSIGN option.

### **LEave**

detaches the tape device without rewinding and unloading it. The tape remains positioned as it was before issuing DETACH. LEAVE is the default if the tape was attached with the NOASSIGN or MULTIUSER option.

## Usage Notes

1. SET IMSG OFF will not suppress informational messages for this command.
2. A previously attached device can be detached from a user, even if the device is still in use.
3. When a device is detached from a user, all Concurrent Copy (CONCOPY) sessions associated with that device for that user are terminated in the storage control.

## DETACH (Real Device)

4. When a device is detached, all data blocks associated with the device are removed from the minidisk cache.
5. If you specify multiple device numbers or a range of device numbers on the DETACH (real device) command line, you receive an affirmative response for each device that is successfully detached and an error message for each device that cannot be detached.
6. If you attempt to detach an active shared device from SYSTEM, you receive an error message indicating that the device is being shared by multiple users. In order to detach the device, each of the affected users must detach the device from their virtual systems.
7. You cannot detach volumes in the CP-owned list that are in use by the system, shared by another system in an SSI cluster, or devices containing minidisks to which users are linked.
8. If a tape device being detached was transferred to a user with the GIVE (Real Device) command, the LEAVE or UNLOAD option specified on the DETACH command overrides any option that was set when the GIVE command was issued. If LEAVE or UNLOAD is not specifically requested on the DETACH command, the option specified when GIVE was issued is processed. If no option was specified on either GIVE or DETACH, the tape is rewound and unloaded.
9. If a device was attached using the CP ATTACH command and later detached, the system operator receives a response saying that the device was detached.
10. For more information, see [“Usage Notes” on page 279](#).
11. When the primary or the secondary volume of a duplex pair is detached from a virtual machine, the response generated will indicate that the volume was either the primary duplex volume or the secondary duplex volume of the duplex pair. When the primary of a duplex pair is detached from the system, the response generated will indicate that the volume was the primary duplex volume of the duplex pair.
12. A tape device attached with the SINGLEUSER option (the default) will automatically be unassigned by CP when the device is detached, unless the LEAVE option is specified on DETACH or the tape was attached with the NOASSIGN option.

A tape device attached with both the SINGLEUSER and NOASSIGN options will not be unassigned by CP when the device is detached, regardless of whether the tape is currently assigned by a guest. In other words, CP will not change the ASSIGNED or UNASSIGNED status of the drive. It is the responsibility of the guest to issue the ASSIGN CCW when it needs the tape drive and to issue the UNASSIGN CCW when it is finished using the drive.
13. A tape device attached with the MULTIUSER option will be unassigned by CP when the device is detached, even though MULTIUSER implies NOASSIGN. This prevents one user from unexpectedly locking out many other users from the drive.
14. When detaching Parallel Access Volume DASDs from a guest, all dedicated alias Parallel Access Volumes associated with a particular base Parallel Access Volume must be detached from that guest before that base Parallel Access Volume can be detached. When detaching Parallel Access Volume DASDs through VOLID, it is going to detach the base and alias associated with the base automatically.
15. When detaching Parallel Access Volume DASDs from the system, all system attached alias Parallel Access Volumes associated with a particular base Parallel Access Volume must be detached from the system before that base Parallel Access Volume can be detached. When detaching Parallel Access Volume DASDs through VOLID, it is going to detach the base and alias associated with the base automatically.
16. When detaching RDEVICES using an EQID, if more than one device exists with the specified EQID, no detach will be performed. To avoid accidental detaches, some other unique identifier (such as the RDEV number) must be used in this situation.
17. When a CP-owned DASD volume with SPOOL extents is detached from the owning system in an SSI cluster, an attempt will be made to automatically detach the volume from all other members in the SSI cluster.

## Responses

The user who currently owns the device (if different from the primary system operator), the user who issues the DETACH (real device) command (if different from the owner of the device), and the primary system operator (if different from the user who issued the command) each receive messages indicating that the DETACH (real device) is successful.

**Note:** The *type* variable in the following responses is omitted when a range of device numbers or multiple device numbers are being reflected in the response.

In the responses below, the term *type* is one of the following:

Type	Meaning
CTCA	Channel-to-channel adapter
CTRL	3745 Communications Controller
DASD	Direct access storage device
FCP	SCSI device adapter
GRAF	Display device
MSC	Mass Storage Controller
OSA	Open Systems Adapter
PRT	Printer
PUN	Card punch
RDR	Card reader
SWCH	Dynamic Switching device
TAPE	Magnetic tape
DEV	Any other device
LINE	Communication line

Response 1:

```
{type vdev} DETACHED BY originator [PRIMARY DUPLEX ]
{vdev...} [SECONDARY DUPLEX]
{vdev-vdev} [PAV BASE ]
[PAV ALIAS ]
[HYPERPAV BASE ]
[HYPERPAV ALIAS ]
```

is the response sent to the device owner (if the device owner is not the primary system operator). Note that the response refers only to the virtual device number (*vdev*) because this is the number by which the owner knows the device.

Response 2:

```
{type rdev} DETACHED {userid [vdev]} [PRIMARY DUPLEX ]
{rdev...} {SYSTEM } [SECONDARY DUPLEX]
{rdev-rdev} [PAV BASE ]
{type ldev} [PAV ALIAS ]
{ldev...} [HYPERPAV BASE ]
{ldev-ldev} [HYPERPAV ALIAS ]
```

is the response issued to the originator of the DETACH (real device) command if the originator is not the owner of the device or devices. When the response is for a single device (*rdev* or *ldev*), the virtual device number, and possibly the duplex state of the device, the PAV state of the device or the HyperPAV state of the device will appear in the responses following the *userid*. Because the secondary volume of a duplex pair cannot be attached to the system, SECONDARY DUPLEX will only be issued when detaching from another user ID.

## DETACH (Real Device)

Response 3:

```
{type rdev} DETACHED {userid [vdev]} BY originator [PRIMARY DUPLEX ]
{rdev...}      {SYSTEM}      {                } [SECONDARY DUPLEX]
{rdev-rdev}   [PAV BASE      ]
{type ldev}   [PAV ALIAS     ]
{ldev...}    [HYPERPAV BASE  ]
{ldev-ldev}  [HYPERPAV ALIAS ]
```

is the response sent to the primary system operator if another user (the originator) issued the DETACH (real device) command. When the response is for a single device (*rdev* or *ldev*), the virtual device number, and possibly the duplex state of the device, the PAV state of the device or the HyperPAV state of the device will appear in the responses following the *userid*. The secondary volume of a duplex pair cannot be attached to the system, therefore SECONDARY DUPLEX will only be issued when detaching from another user ID.

Response 4:

```
{TAPE rdev} DETACHED userid BY originator {, TAPE VOLUME(S) UNLOADED }
{rdev...}   {                } { TAPE VOLUME(S) NOT UNLOADED}
{rdev-rdev}
```

is the response issued to the system operator when one or more tape volumes are being detached.

Response 5:

```
TAPE vdev RETURNED
```

is the response the user receives when a tape drive given to another user through the GIVE command is returned.

Response 6:

```
TAPE rdev RETURNED TO userid vdev
```

is the response the primary system operator receives when a tape drive given to another user through the GIVE command is returned.

Response 7:

```
Detaching vdev has caused the termination of
CONCOPY session(s) on device(s): vdev1...
```

is the response the device owner receives if the process of detaching the device causes the termination of Concurrent Copy (CONCOPY) sessions, and the device owner is the originator of the DETACH command. The response lists the virtual devices that are part of the terminated CONCOPY sessions.

Response 8:

```
Detaching vdev by userid has caused the termination
of CONCOPY session(s) on device(s): vdev1...
```

is the response the device owner receives if the process of detaching the device causes the termination of Concurrent Copy (CONCOPY) sessions, and the device owner is *not* the originator of the DETACH command. The response identifies the originator of the DETACH command and lists the virtual devices that are part of the terminated CONCOPY sessions.

Response 9:

```
Detaching rdev has caused the termination of
CONCOPY session(s) for userid userid
```

is the response the originator of the DETACH command receives if the process of detaching the device causes the termination of Concurrent Copy (CONCOPY) sessions, and the originator of the command is *not* the owner of the device. The response identifies the user whose CONCOPY sessions have been terminated.

Response 10:

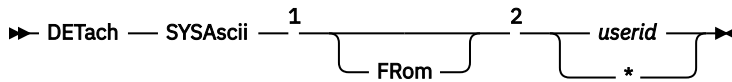
```
Detaching rdev by userid1 has caused the termination  
of CONCOPY session(s) for userid userid2
```

is the response the primary system operator receives if the process of detaching the device from user *userid2* causes the termination of Concurrent Copy (CONCOPY) sessions, and the system operator is *not* the originator of the DETACH command (*userid1*).

### Messages

- HCP048E No device with EQID *eqid* attached.
- HCP048E No device with EQID *eqid* exists.
- HCP135E Multiple devices with EQID *eqid* attached.
- HCP135E Multiple devices with EQID *eqid* exist.

## DETACH SYSASCII



### Notes:

- <sup>1</sup> Operands are valid only for class B users.
- <sup>2</sup> If DETACH SYSASCII is executed with the AT command, the *userid* operand is required.

## Authorization

Privilege Class: B, G

## Purpose

Use DETACH SYSASCII to remove the real ASCII console from a virtual machine configuration. If you are a class G user, you can remove the ASCII console from your own virtual machine. If you are a class B user, you can remove the ASCII console from any virtual machine.

## Operands

### *userid*

identifies the virtual machine to which the ASCII console is currently attached. The specified user must be logged on and cannot be in the process of logging on.

\*

detaches the ASCII console from the user who issues the command.

## Usage Notes

1. SET IMMSG OFF does not suppress informational messages for this command.
2. The ASCII console can be detached from a user even if it is still in use.
3. To avoid inadvertent transfer of data between virtual machines through the ASCII console, the screen is erased when the device is detached.
4. Whenever the ASCII console is detached, the system operator receives a message indicating that the facility is available.

## Responses

Receiving messages indicating that the DETACH SYSASCII was successful are:

- The user who currently owns the ASCII console
- The user who issued the DETACH SYSASCII command (if different from the owner of the ASCII console)
- The primary system operator (if different from the user who issued the command).

Response 1:

```
SYSASCII detached by originator
```

is the response sent to the owner of the ASCII console.

Response 2:

```
SYSASCII detached from userid
```

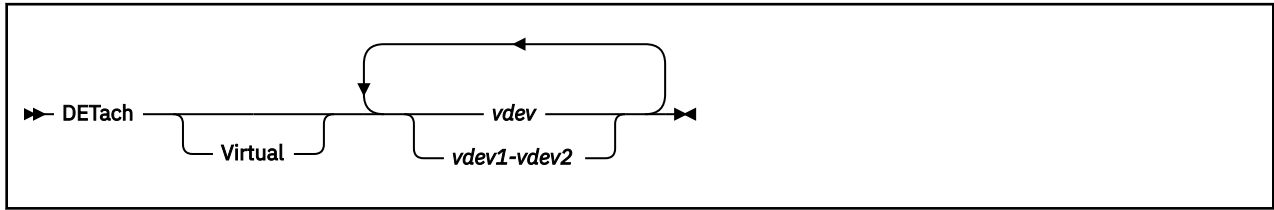
is the response issued to the originator of the DETACH SYSASCII command if the originator is not the owner of the ASCII console.

Response 3:

```
SYSASCII detached from userid by originator
```

is the response sent to the primary system operator when another user (identified by *originator*) issues DETACH SYSASCII.

## DETACH (Virtual Device)



### Authorization

Privilege Class: G

### Purpose

Use DETACH (virtual device) to remove devices from your virtual machine configuration.

### Operands

*vdev*

*vdev1-vdev2*

is a virtual device number, a list of virtual device numbers, or a range of virtual device numbers to be detached from your virtual machine. On a single command line, you may combine device numbers, lists of device numbers, and ranges of device numbers. If you are detaching a list of virtual devices, insert at least one blank between the device numbers you specify. If you are detaching a range of devices, insert a hyphen (-) between the device numbers you specify. The maximum number of devices allowed within a range is 256.

### Usage Notes

1. You can detach a virtual device even if it is currently in use.
2. When you detach a dedicated device, it becomes inaccessible to your virtual machine; it is released and becomes available to any virtual machine, another user, or the CP system. The system operator receives a response indicating that the device is free.
3. If the virtual device is part of a Concurrent Copy (CONCOPY) session for your virtual machine, the CONCOPY session is terminated in the storage control associated with the device.
4. Tape devices are automatically rewound and unloaded when you enter the DETACH command, unless the tape device was attached with the NOASSIGN or MULTIUSER option.
5. If you detach a dedicated device, and it was not obtained using a GIVE command with return specified, the system operator receives a message indicating that the device is free.
6. Command processing does not end if an error (other than a syntax error) is encountered when processing a range of device numbers or multiple device numbers. Instead, an error message is issued for each device that cannot be detached. Informational messages are issued to identify those devices that were successfully detached. These informational messages may appear several times in one command execution when you have specified on the command line a range of device numbers or multiple device numbers.
7. A tape device attached with the SINGLEUSER option (the default) will automatically be unassigned by CP when the device is detached, unless the device was attached with the NOASSIGN option.

A tape device attached with both the SINGLEUSER and NOASSIGN options will not be unassigned by CP when the device is detached, regardless of whether the tape is currently assigned by a guest. In other words, CP will not change the ASSIGNED or UNASSIGNED status of the drive. It is the responsibility of the guest to issue the ASSIGN CCW when it needs the tape drive and to issue the UNASSIGN CCW when it is finished using the drive.



8. A tape device attached with the MULTIUSER option will be unassigned by CP when the device is detached, even though MULTIUSER implies NOASSIGN. This prevents one user from unexpectedly locking out many other users from the drive.
9. Devices that are part of a virtual network adapter cannot be detached individually. Virtual network adapters are detached with the DETACH NIC command.
10. If you use DETACH to detach a virtual disk in storage, and you are the last user, the virtual disk in storage is destroyed.
11. SET IMSG OFF suppresses informational messages for this command.
12. When detaching Parallel Access Volume DASDs, all alias Parallel Access Volumes associated with a particular base Parallel Access Volume must be detached first before that base Parallel Access Volume can be detached.
13. This command will fail if issued for a VDISK or virtual disk in storage when VMRELOCATE is in progress.
14. For more information, see [“Usage Notes” on page 279](#).

## Responses

In Response 1, *type* is one of the following:

Type	Meaning
CONS	Console
CTCA	Channel-to-channel adapter
CTLR	3745 Communications Controller
DASD	Direct access storage device
DEV	Any other device
FCP	SCSI device adapter
GRAF	Display device
MSC	Mass Storage Controller
OSA	Open Systems Adapter
PRT	Printer
PUN	Card punch
RDR	Card reader
SWCH	Dynamic Switching device
TAPE	Magnetic tape

The *type* variable in the remaining responses is omitted when a range of device numbers or multiple device numbers are being reflected in the response.

Response 1:

```
{type vdev} DETACHED
{vdev ... }
{vdev-vdev}
```

is the response you receive for the device or devices that were successfully detached (if you are not the primary system operator).

Response 2:

```
CTCA vdev DROPPED FROM userid vdev
```

## DETACH (Virtual Device)

is the response if the device detached was a virtual CTCA connected using the COUPLE command.

### **CTCA *vdev***

is the virtual device number of the CTCA that you detached.

### ***userid vdev***

is the user identification and virtual device number of the CTCA that your CTCA had been coupled to.

This response is always followed by the response:

```
CTCA vdev DETACHED
```

to indicate that the command has been successfully processed. The response:

```
CTCA vdev DROPPED BY userid vdev
```

is sent to the user whose CTCA connection was dropped.

Response 3:

```
Tape vdev returned
```

is the response sent to a user who had issued the GIVE command with the RETURN option. The response is sent when the receiver of the tape detaches the tape.

### ***vdev***

is the virtual device number of the tape drive that was returned.

Response 4:

```
Tape rdev returned to userid vdev
```

is the response the primary system operator receives when a tape drive given to another user through the GIVE command is returned.

### ***rdev***

is the real device number of the tape drive.

### ***userid***

is the user ID the tape drive is being returned to.

### ***vdev***

is the virtual device number of the tape drive that was returned.

Response 5:

```
Detaching vdev has caused the termination of  
CONCOPY session(s) on device(s): vdev1...
```

indicates that one or more Concurrent Copy (CONCOPY) sessions have been terminated during the process of detaching the device.

### ***vdev***

is the virtual device number that you are detaching.

### ***vdev...***

is a list of the device numbers that are part of the terminated CONCOPY sessions.

## DETACH VSWITCH

```
▶▶ DETach VSWITCH — switchname ▶▶
```

### Authorization

Privilege Class: B

### Purpose

Use the DETACH VSWITCH command to remove the designated virtual switch from the CP system, uncouple any virtual adapters using the virtual switch, and release system resources associated with it.

### Operands

#### *switchname*

is the name of the virtual switch. The *switchname* is a single token of 1–8 alphanumeric characters.

### Usage Notes

1. Only a Class B user can DETACH a virtual switch.
2. Any virtual network adapter coupled to the virtual switch receives an indication that the LAN has stopped functioning.

### Examples

1. To detach virtual switch BIGANG from the system, enter the following:

```
detach vswitch bigang
```

### Responses

Response 1:

```
VSWITCH SYSTEM switchname is destroyed
```

This is the normal response for this command.

Response 2:

```
NIC vdev disconnected from VSWITCH SYSTEM switchname
```

This is the response when a virtual network adapter connected to a virtual switch is detached.

### Messages

- HCP3166E [ DETACH VSWITCH | SET VSWITCH DISCONNECT ] for IVL VSWITCH *ivlswitchnm* failed. Virtual switch *switchname* is defined on *systemnm* with the GLOBAL attribute.

## DIAL



### Authorization

Privilege Class: Any

### Purpose

Use DIAL to logically connect your terminal to a virtual machine.

### Operands

#### *userid*

identifies the virtual machine to which you want to connect. SYSTEM and asterisk (\*) are not valid user IDs for this command.

#### *vdev*

identifies the device number of the virtual terminal to which the connection is to be made.

### Usage Notes

1. The system accepts a DIAL command only at logon, and only as a substitute for a LOGON command.
2. Once the connection is made, your terminal operates entirely under the control of the virtual machine. The DIAL command matches your terminal to an equivalent type defined in the virtual machine. If no matching terminal type exists, the connection cannot be made, and an error message is issued.
3. The first-level operator of a virtual system can return the terminal to CP control by issuing either the RESET command or the DETACH command. A virtual system user can return the terminal by causing an unsolicited device end to be generated for the dialed terminal. To generate an unsolicited device end, use the TEST/NORMAL or POWER OFF/POWER ON functions.
4. A SNA/CCS terminal user cannot dial into the VTAM service machine controlling the terminal. A logical terminal user cannot dial into the virtual machine that created it.
5. The write-for-position operation is not supported for SNA/CCS terminals that are controlled by a level of VTAM service machine that is prior to ACF/VTAM version 3, release 1.1, for VM/SP (VM/VTAM). For this operation, a write CCW is chained to a read CCW; the write operation sets the 3270 buffer address from which the read operation is to begin. If a user attempts a write-for-position operation that is not supported by the controlling VTAM service machine, the virtual machine receives an operation exception on the read operation.
6. The DIAL command is not supported for SNA/CCS terminals controlled by a VCNA level of VTAM service machine. ACF/VTAM version 3 (VM/VTAM) VSMs, and later, support the DIAL command for SNA/CCS displays. The DIAL command is not supported for ASCII terminals attached through SNA/CCS.
7. If an external security manager is installed on your system, the DIAL command may be disabled. For additional information, contact your security administrator.

### Responses

Response 1:

```
DIALED TO userid vdev
```

is the response sent to the user indicating that a logical connection has been established.

**userid**

identifies the virtual machine in which the multiple-access operating system that you want to use is running.

**vdev**

identifies the virtual device number of the terminal to which the dialed connection has been made.

Response 2:

```
{type {rdev} } DIALED TO userid DIALED = nnnnn [FROM ipadr]
{
  {ldev}
}
{luname }
```

is the response sent to the primary system operator. It indicates a successful connection to the virtual machine (*userid*) and the total number of terminals (*nnnnn*) currently connected to other virtual machines.

**type**

identifies the user's terminal device type.

**rdev**

**ldev**

identifies the real or logical device number of your console.

**luname**

identifies a SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

**FROM ipadr**

identifies the IP address.

An IPv4 address is displayed in dotted-decimal format.

**Example:** 9 . 130 . 44 . 144

An IPv6 address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros).

**Example:** The following is an IPv6 address:

```
F802:0000:0000:0000:0000:0000:0A00:0001
```

This address is displayed in compressed format as:

```
F802::A00:1
```

Response 3:

```
DROP FROM userid vdev
```

is the message sent to the user when the display is reset.

The system additionally responds with one of the following, depending on the type of terminal being used:

Type of Terminal	Message
3270 family	Press the ENTER or CLEAR key to continue
SNA/CCS 3270	Press the ENTER key to continue

Pressing the appropriate key on your terminal causes the logo to appear on the terminal, which readies the terminal for the next session. For terminals not listed above, control automatically returns to CP, and the "online" message is printed.

Response 4:

```
{type {rdev} } DROP FROM userid DIALED = nnnnn
{
  {ldev}
}
{luname }
```

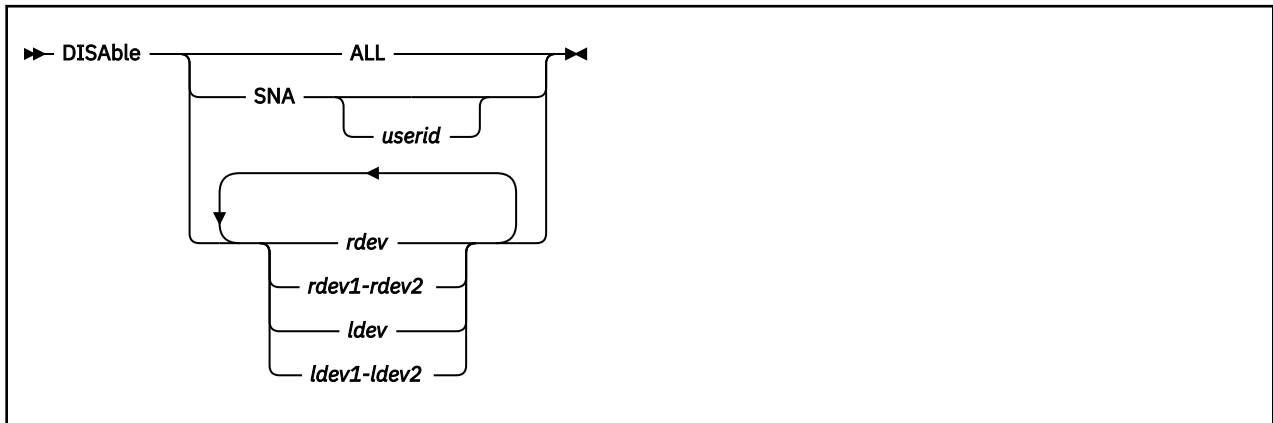
## DIAL

is the message sent to the primary system operator when the display is reset.

### Messages

- HCP011E Invalid device type - *userid vdev*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP045E *userid* not logged on
- HCP047E *userid vdev* does not exist
- HCP055E Line(s) not available on *userid*
- HCP056E Line *vdev* busy on *userid*
- HCP206E Cannot connect to host virtual machine
- HCP743I DIAL function not available

## DISABLE (Device)



### Authorization

Privilege Class: A, B, F

### Purpose

Use DISABLE (device) to prevent the following devices from accessing the host system:

- Display devices
- EP (Emulator Program) controlled lines in the 3745
- SNA/CCS terminals.

### Operands

#### ALL

disables all real 3270 displays, including the operator's display, all real 328x printers, EP (Emulator Program) controlled lines in the 3745, and all SNA/CCS terminals.

**Note:** The logical 328x printers are not disabled.

#### SNA *userid*

disables communication with the SNA/CCS terminals owned by the VTAM service machine whose user ID is specified. If *userid* is omitted, communication with all SNA/CCS terminals for all VTAM service machines is disabled.

#### *rdev*

#### *rdev1-rdev2*

is a real device number, a list of real device numbers, or a range of real device numbers of display stations or lines that you want to disable. Specify each real device number as 1- to 4-hexadecimal digits. If you specify more than one device number, each must be separated from the others by at least one blank.

#### *ldev*

#### *ldev1-ldev2*

is a logical device number, a list of logical device numbers, or a range of logical device numbers of 328x display printers that you want to disable. Only 328x display printers are valid for this operand. Specify a logical device number as an L followed by 4 hexadecimal digits. If you specify more than one device number, each must be separated from the others by at least one blank.

### Usage Notes

1. On a single command line you may combine device numbers, lists of device numbers, and ranges of device numbers. The beginning and ending device numbers of a range must both be either real device numbers or both logical display printer numbers.
2. If the devices you specify on the DISABLE (device) command are not active, they are disabled immediately. If the devices are active, they are disabled when the user issues the LOGOFF or DISCONNECT command. A device is active if a user is logged onto it, or if a user is dialed from it.
3. If you enter the ENABLE (device) command to a device before it is physically disabled, you reinstate the device's enabled status and negate previously issued DISABLE commands directed to the device.
4. The DISABLE (device) command is an ASYNCHRONOUS command. This means the virtual machine is free to issue other commands before CP finishes disabling the specified devices. CP issues the "Command complete" response when it has finished processing the DISABLE (device) command, not when it has finished disabling all of the specified devices. Thus, one or more devices may not be completely disabled when you receive the "Command complete" response.

To display whether the specified real devices (*rdev*) are disabled, use the QUERY (Real Device) STATUS command (see [“QUERY \(Real Device\)”](#) on page 1114). To display whether the specified logical devices (*ldev*) are disabled, use the QUERY LDEVS command (see [“QUERY LDEVS”](#) on page 973).

### Responses

```
Command complete
```

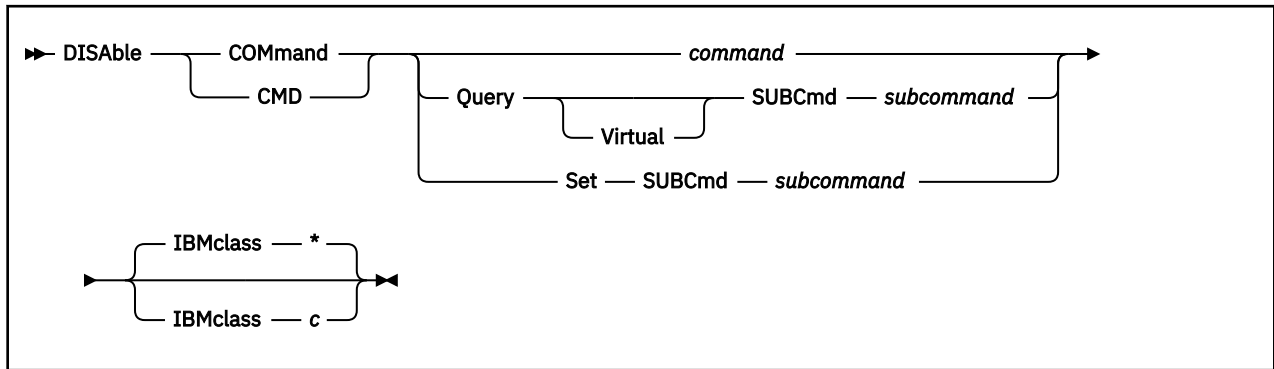
is issued when the command completes successfully.

### Messages

- HCP002E Invalid operand - *operand*
- HCP006E Invalid device type - {*rdev* | *vdev* | *ldev*}
- HCP009E Invalid range - *range*
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev* | *vdev* | *ldev*} does not exist
- HCP046E *devtype* *rdev* offline
- HCP053E [XAUTOLOG failed for *userid:*] {*userid* | *value*} not in CP directory
- HCP140E *devtype* {*rdev* | *ldev*} attached to *userid*
- HCP263E Too many operands were supplied for this command.



## DISABLE COMMAND / CMD



### Authorization

Privilege Class: A

### Purpose

Use DISABLE COMMAND or CMD to prevent CP from processing requests for the specified CP command.

You can also prevent processing of CP commands during initialization using the DISABLE COMMAND or CMD configuration file statements. For more information, see [DISABLE COMMAND / CMD Statement in z/VM: CP Planning and Administration](#).

### Operands

#### **command**

is the name of the command that you are disabling. The variable *command* is a 1-character to 12-character alphanumeric string.

#### **Query SUBCmd subcommand**

tells CP the name of the CP QUERY subcommand that you are disabling. The variable *subcommand* is a 1-character to 12-character alphanumeric string.

#### **Query Virtual SUBCmd subcommand**

tells CP the name of the CP QUERY VIRTUAL subcommand that you are disabling. The variable *subcommand* is a 1-character to 12-character alphanumeric string.

#### **Set SUBCmd subcommand**

tells CP the name of the CP SET subcommand that you are disabling. The variable *subcommand* is a 1-character to 12-character alphanumeric string.

#### **IBMclass \***

tells CP to disable all versions of the specified command or subcommand. If omitted, IBMCLASS \* is the default.

#### **IBMclass c**

tells CP to disable a specific version of the specified command or subcommand. The variable *c* can be any one of the following:

##### **A**

this is a system-control command to be used by the primary system operator.

##### **B**

this is a command for operational control of real devices.

##### **C**

this is a command to alter system storage.

### D

this is a command for system-wide control of spool files.

### E

this is a command to examine system storage.

### F

this is a command for service control of real devices.

### G

this is a general-use command used to control the functions of a virtual machine.

### O

(zero) this command has no specific IBM class assigned.

## Usage Notes

1. To remove the command processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD” on page 117](#)).
2. To load the command processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD” on page 111](#)).
3. To deactivate a CP command while defining it, use the DISABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD” on page 143](#)).

If you disable a command that has an alias defined, then the alias will be treated as if it were also disabled.

4. To activate a CP command:
  - While defining it, use the ENABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD” on page 143](#)).
  - After defining it, use the ENABLE COMMAND or CMD command (see [“ENABLE COMMAND / CMD” on page 460](#)).
5. To change the definition of an existing CP command, use the MODIFY COMMAND or CMD command (see [“MODIFY COMMAND / CMD” on page 683](#)).
6. Once defined, commands, subcommands, and aliases cannot be deleted. They may be altered in various appropriate ways, but they remain in existence until a SHUTDOWN or RESTART IPL is done.
7. To obtain information such as the enable/disable status and the name of the base command for a command or alias that is defined, use the QUERY CPCMDS command (see [“QUERY CPCMDS” on page 817](#)).
8. To display the address of the CP command table entry block, the current IBM class, and the current privilege class for a specified CP command, use the LOCATE CMDBK command (see [“LOCATE CMDBK” on page 603](#)).
9. For more information about enabling and disabling commands, see [z/VM: CP Exit Customization](#).

## Responses

### Response 1:

To disable the class D version of the CP PURGE command, but not the class G version of the CP PURGE command, enter the following:

```
disable command purge ibmclass d
Ready;
```

### Response 2:

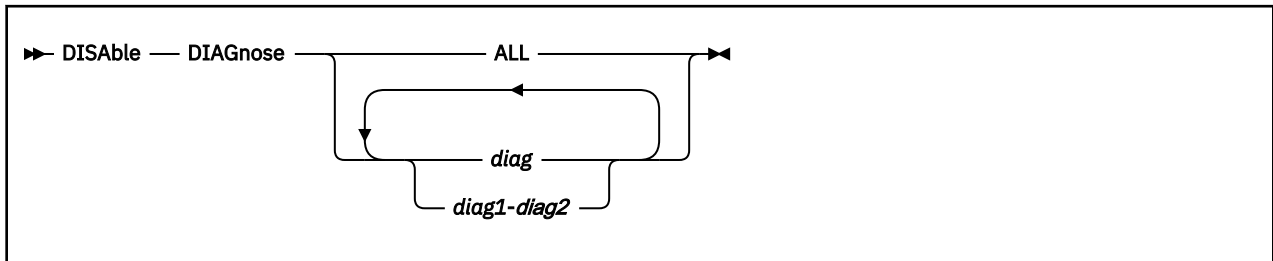
To disable the CP SHUTDOWN command, enter the following:

```
disable command shutdown  
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid command name - *command*
- HCP6706E Invalid IBM class - *c*
- HCP6706E Invalid privilege classes - *classes*
- HCP6706E Invalid string - *string*
- HCP6706E Invalid subcommand name - *subcommand*
- HCP6840E You specified QUERY VIRTUAL without SUBCMD
- HCP6840E You specified SUBCMD without SET or QUERY
- HCP6841E You did not specify an IBM class.
- HCP6842E Cannot find existing CP command
- HCP9300E A system soft abend *code* has occurred on CPU *nnnn* for user *userid*.

## DISABLE DIAGNOSE



### Authorization

Privilege Class: A

### Purpose

Use `DISABLE DIAGNOSE` to prevent CP from processing requests for one or more locally-developed `DIAGNOSE` codes.

You can also prevent processing of locally-developed `DIAGNOSE` codes during initialization using the `DISABLE DIAGNOSE` configuration file statement. For more information, see [DISABLE DIAGNOSE Statement](#) in *z/VM: CP Planning and Administration*.

### Operands

#### ALL

tells CP to disable all existing `DIAGNOSE` codes.

**Note:** This operand disables all `DIAGNOSE` codes: the locally-defined ones, the IBM-supplied ones, and any supplied by third-party software vendors.

#### *diag*

#### *diag1-diag2*

is the number of the `DIAGNOSE` code that you are disabling. Each *diag* must be a hexadecimal number between X'0000' and X'03FC' and must be a multiple of 4. You can specify a single `DIAGNOSE` code, a range of `DIAGNOSE` codes, or any combination thereof.

### Usage Notes

1. If a `DIAGNOSE` code that you are disabling is currently processing a `DIAGNOSE` request, CP waits until the current request has finished before disabling the `DIAGNOSE` code. If there are `DIAGNOSE` requests waiting to execute after the current request, CP waits until the current request has finished, disables the `DIAGNOSE` code, and rejects any `DIAGNOSE` requests still waiting to be processed.
2. CP treats disabled `DIAGNOSE` codes as if they were never defined. If you try to use a disabled `DIAGNOSE` code in a program, CP will give you a program check specification exception.
3. To load the `DIAGNOSE` processing code into the system execution space, use the `CPXLOAD` command (see “[CPXLOAD](#)” on page 111).
4. To define a new `DIAGNOSE` code, use the `DEFINE DIAGNOSE` command (see “[DEFINE DIAGNOSE](#)” on page 172).

**Note:** Unless you specify the `ENABLE` operand of the `DEFINE DIAGNOSE` command, the new `DIAGNOSE` code is initially in a disabled state after being defined.

5. To activate a new `DIAGNOSE` code after defining it, use the `ENABLE DIAGNOSE` command (see “[ENABLE DIAGNOSE](#)” on page 463).

6. To change the definition of an existing DIAGNOSE code after initialization, use the MODIFY DIAGNOSE command (see [“MODIFY DIAGNOSE”](#) on page 697).
7. Once defined, a DIAGNOSE code cannot be deleted. It may be altered in various appropriate ways, but it remains in existence until a SHUTDOWN or RESTART IPL is done.
8. To display information about a DIAGNOSE code (status, entry point name, and privilege class), use the QUERY DIAGNOSE command (see [“QUERY DIAGNOSE”](#) on page 885).
9. To display the address of the CP DIAGNOSE code table block for a DIAGNOSE code, use the LOCATE DGNBK command (see [“LOCATE DGNBK”](#) on page 608).
10. To remove the DIAGNOSE processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117).
11. Many external security managers (ESMs) do not support DIAGNOSE codes above X'03FC'. For this reason, CP does not support DIAGNOSE codes above X'03FC'. The DIAGNOSE codes between X'0000' and X'03FC' are divided as follows:
  - X'0000' to X'00FC'**  
Reserved for IBM use
  - X'0100' to X'01FC'**  
Reserved for customer use
  - X'0200' to X'03FC'**  
Reserved for IBM use.
12. For more information about user-defined DIAGNOSE codes, see [Defining and Modifying Commands and Diagnose Codes in z/VM: CP Exit Customization](#).

## Responses

Response 1 (Disable One):

To disable DIAGNOSE code X'100', enter the following:

```
disable diagnose 100
Ready;
```

After CP finishes processing this command, any virtual machine that issues DIAGNOSE code X'100' receives a program specification exception in that virtual machine.

Response 2 (Disable All):

To disable all DIAGNOSE codes on your system, enter the following:

```
disable diagnose all
Ready;
```

Response 3 (Disable All Local):

To disable all locally-defined DIAGNOSE codes, enter the following:

```
disable diagnose 100-1fc
Ready;
```

Response 4 (Disable Almost All Local):

To disable all locally-defined DIAGNOSE codes except DIAGNOSE code X'01F0', enter the following:

```
disable diagnose 100-iec if4-1fc
Ready;
```

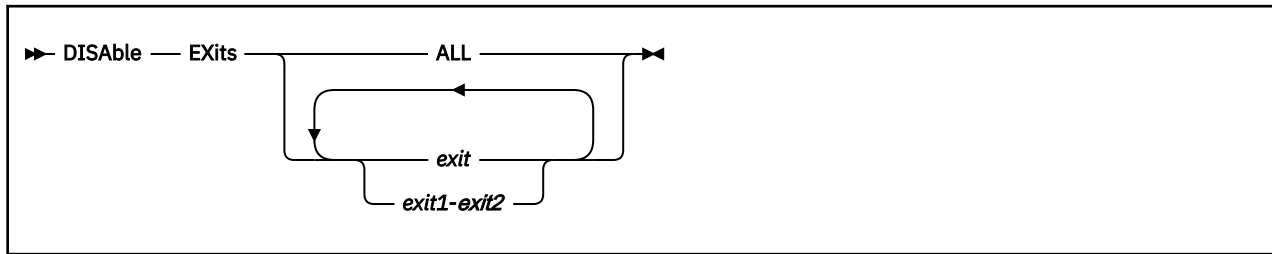
## Messages

- HCP002E Invalid operand - *operand*

## DISABLE DIAGNOSE

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP2751E Diagnose code *diag* does not exist
- HCP2751E Diagnose codes *diag1-dia2* do not exist
- HCP6704E Missing token at end of line
- HCP6706E Invalid Diagnose code - *diag*

## DISABLE EXITS



### Authorization

Privilege Class: A

### Purpose

Use `DISABLE EXITS` to prevent CP from calling all entry points and external symbols associated with one or more exit points.

You can also prevent CP from calling one or more exit points during initialization using the `DISABLE EXITS` configuration file statement. For more information, see [DISABLE EXITS Statement](#) in *z/VM: CP Planning and Administration*.

### Operands

#### ALL

tells CP to disable all existing CP exit points.

#### exit

#### exit1-exit2

is the number of the exit point (or exit points) that you no longer want CP to use. Each *exit* must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single exit point number, a range of exit point numbers, or any combination thereof.

### Usage Notes

1. If the exit point is currently processing an exit request, CP waits until that exit request is completed and then disables the exit point. If there are more exit requests waiting to be processed, CP places your `DISABLE EXIT` command in the exit request queue behind all the known exit requests. After processing all the known exit requests, CP disables the specified exit point and rejects any exit requests that arrived after it placed your `DISABLE EXITS` command in the exit request queue.
2. To load the exit point code into the system execution space, use the `CPXLOAD` command (see [“CPXLOAD”](#) on page 111).
3. To add to, change, or replace the list of entry points and external symbols associated with an exit point and to enable or disable that exit point, use the `ASSOCIATE EXITS` command (see [“ASSOCIATE EXIT”](#) on page 20).
4. To activate an exit point after associating it, use the `ENABLE EXITS` command (see [“ENABLE EXITS”](#) on page 465).
5. To display status and usage statistics information about a specific exit point, use the `QUERY EXITS` command (see [“QUERY EXITS”](#) on page 907).
6. To display the address of the CP exit block for a specific exit point, use the `LOCATE XITBK` command (see [“LOCATE XITBK”](#) on page 649).
7. To display the address of the CP indirect call locator block for a specific exit point, use the `LOCATE ICLBK` command (see [“LOCATE ICLBK”](#) on page 614).

## DISABLE EXITS

8. To change the definition of an existing dynamic exit point, or to remove the exit point from the system, use the MODIFY EXIT command (see [“MODIFY EXIT” on page 700](#)).
9. To remove the exit point code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD” on page 117](#)).
10. After processing a DISABLE EXITS command (or statement), CP updates the status of the exit point in its CP exit block, but does not erase the CP exit block. CP will not erase any CP exit blocks until the next IPL.
11. For more information about user-defined exit points, see [z/VM: CP Exit Customization](#).

## Responses

Response 1:

To stop CP from calling the entry points and external symbols associated with CP Exits 1, 2, 3, 4, and 6, enter the following:

```
disable exits 1-4 6
Ready;
```

Response 2:

To stop CP from calling the entry points and external symbols associated with all CP exit points, enter the following:

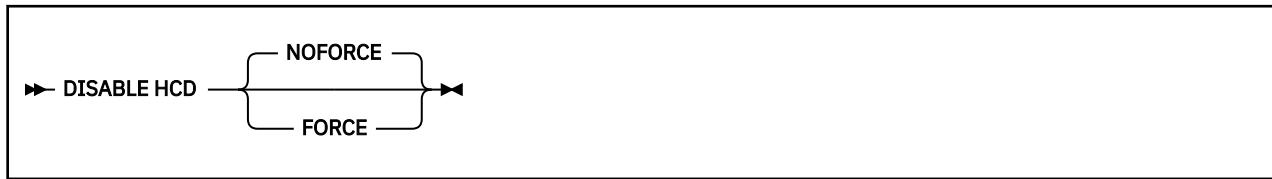
```
disable exits all
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP2752E Exit number *exit* does not exist
- HCP2752E Exit numbers do not exist
- HCP2752E Exit numbers *exit1-exit2* do not exist
- HCP6704E Missing token at end of line
- HCP6706E Invalid CP Exit number - *exit*



## DISABLE HCD



### Authorization

Privilege Class: B

### Purpose

Use the DISAbLe HCD command to disable the I/O-configuration capabilities of HCM and HCD for the rest of the current IML of your machine, transferring I/O configuration control to the applicable CP operator commands.

With HCM and HCD disabled, the hardware I/O configuration can be changed with the CP dynamic-I/O-configuration commands SET DYNAmic\_i/o, SET CONFIGmode, DEFine CHPID, DEFine CNTLUNIT, DEFine DEVEice, MODify CHPID, MODify CNTLUNIT, MODify DEVEice, DELete CHPID, DELete CNTLUNIT, DELete DEVEice, Set IOCDs\_active, and Set TOKen.

Additionally, with HCM and HCD disabled, the VM software I/O configuration can be changed with the CP operator commands Set RDEVEice, DELete RDEVEice, Set EDEVEice, DELete EDEVEice, Set DEVEices, and Set HOTIO.

### Operands

#### NOFORCE

can be specified to cause the command to be rejected if, as part of disabling HCM and HCD and enabling the CP dynamic-I/O-configuration commands, a VM configuration token cannot be established. The inability to establish a VM configuration token in the machine means that the CP dynamic-I/O-configuration commands cannot be used. Therefore the NOFORCE operand leaves the HCM and HCD capability intact if the CP dynamic-I/O-configuration commands cannot be made effective.

#### FORCE

can be specified to force execution of the command even though a VM configuration token cannot be established. In this case HCM and HCD are disabled from making I/O-configuration changes, and in addition the VM dynamic-I/O-configuration commands remain disabled. No hardware I/O-configuration changes are possible without a re-IML of your machine (thereby taking down all logical partitions if the machine is logically partitioned). Software I/O-configuration changes can still be made, however, using the applicable CP operator commands.

### Usage Notes

1. Once your machine is IMLed with HCM and HCD in control of the I/O configuration, the DISAbLe HCD command is a one-way escape mechanism to take control away from HCM and HCD and attempt to give it to the CP operator commands.

Note that once HCM and HCD are disabled, there is no way to re-enable them without re-IMLing your machine (thereby taking down all logical partitions if the machine is logically partitioned).

2. If HCM and HCD are currently disabled for hardware dynamic-I/O-configuration, this command does not attempt to establish a VM configuration token in the machine. No attempt is made to change the token (which would allow the use of the CP-dynamic-I/O-configuration commands) because “ownership” of the existing configuration token is uncertain and it is not known whether HCM and

## DISABLE HCD

HCD were ever used to dynamically change the machine configuration. In this case, whether *FORCE* or *NOFORCE* is specified, the command is executed and HCM and HCD are disabled. Even though the machine I/O configuration cannot be changed by any means in this case without a re-IML of your machine,, there is still the capability to dynamically change the software I/O configuration using CP operator commands.

3. If HCM and HCD are currently enabled for hardware dynamic-I/O-configuration when this command is issued and the configuration token is successfully changed to a VM configuration token, the SET DYNamic\_i/o ON command can be issued to enable the use of the CP dynamic-I/O-configuration commands.

### Responses

#### Response 1:

When the DISAbLe HCD command is issued and the VM/HCD service virtual machine does not exist or is not being used to control the I/O configuration, the following response will be issued:

```
disable hcd
HCD is not active
Ready;
```

#### Response 2:

When the DISAbLe HCD command is issued and the VM/HCD service virtual machine is enabled for hardware I/O-configuration changes, the following response is issued when the configuration token is successfully changed to a VM token in the machine:

```
disable hcd force
HCD was successfully disabled
Ready;
```

#### Response 3:

When the DISAbLe HCD command is issued and the VM/HCD service virtual machine is not enabled for hardware I/O-configuration changes, the following informational-message and command-response combination is issued:

```
disable hcd
HCPCCY654302I HCD is now disabled, but a VM configuration token
HCPCCY654302I was not established because HCD was not enabled for
HCPCCY654302I dynamic hardware changes.
Command complete.
Ready;
```

#### Response 4:

When the DISAbLe HCD FORCE command is issued and the VM/HCD service virtual machine is enabled for hardware I/O-configuration changes, the following informational-message and command-response combination is issued if the machine will not enter configuration mode to change the current configuration token to a VM configuration token:

```
disable hcd force
HCPCCY654301I HCD is now disabled, but the machine would not enter
HCPCCY654301I configuration mode to allow a VM configuration token
HCPCCY654301I to be established.
Command complete.
Ready;
```

#### Response 5:

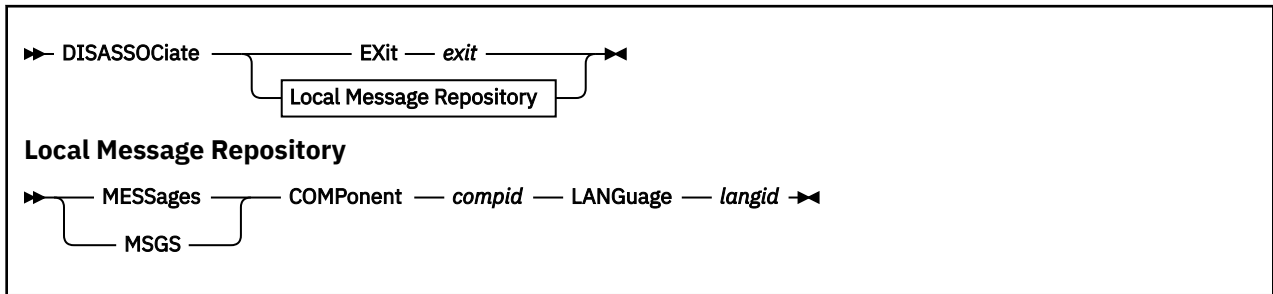
When the DISAbLe HCD command is issued and CP is in the process of updating the I/O configuration for the VM/HCD service virtual machine, the following response will be issued:

```
disable hcd
HCD is currently updating the I/O configuration and cannot be disabled.
Ready;
```

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6544E Disable HCD command failed. The machine would not enter configuration mode to allow a VM configuration token to be established.

## DISASSOCIATE



### Authorization

Privilege Class: A

### Purpose

Use DISASSOCIATE to revoke all entry point and external symbol assignments associated with an exit point or local message repository using ASSOCIATE EXIT, MESSAGES, or MSGS commands or configuration file statements.

### Operands

#### EXIT *exit*

tells CP that you want to disassociate all entry points and external symbols from the specified exit point. The variable *exit* must be a hexadecimal number between X'0000' and X'FFFF'.

#### MESSAGES COMPONENT *compid* LANGUAGE *langid*

#### MSGS COMPONENT *compid* LANGUAGE *langid*

tells CP that you want to disassociate all entry points and external symbols from the specified local message repository.

The variable *compid* is a 1- to 3-character alphanumeric string. For example, the component ID for the system message repository (z/VM) is HCP.

The variable *langid* is a 1- to 5-character alphanumeric string. AMENG (American English) and UCENG (Uppercase English) are examples of two very common language IDs.

You can use generic language names to request a specific subset of languages. A generic language name is a 1- to 5-character string with asterisks (\*) in place of 1 or more characters and percent signs (%) in place of exactly 1 character. For example:

```
... language %%en*
```

disassociates all languages that have EN as their third and fourth characters.

### Usage Notes

1. When you use the DISASSOCIATE command for an exit point, CP does not change the status (enabled or disabled) of the exit point. If the exit point is enabled, it will remain enabled after your DISASSOCIATE command completes. If the exit point is disabled, it will remain disabled after your DISASSOCIATE command completes. Only the ASSOCIATE EXIT, DISABLE EXITS, and ENABLE EXITS commands (or configuration file statements) can change the status of the exit point.

To avoid unnecessary system processing, you should disable the exit point either before or after you issue the DISASSOCIATE command. If you do not disable the exit point, the effect is the same as having a single entry point name that did nothing except send back a return code of 0 (zero) to the mainline code.

2. After you revoke all entry point and external symbol assignments using the DISASSOCIATE command, you need to unload the customer-written CP routines using the CPXUNLOAD command (see [“CPXUNLOAD” on page 117](#)).
3. To assign entry points and external symbols to an exit point and to enable or disable that exit point, use the ASSOCIATE EXIT command (see [“ASSOCIATE EXIT” on page 20](#)).
4. To assign an external symbol to a local message repository, use the ASSOCIATE MESSAGES or MSGS command (see [“ASSOCIATE MESSAGES / MSGS” on page 24](#)).
5. After issuing DISASSOCIATE EXIT, CP erases the entry point names and external symbols from the CP exit block for that exit point, but CP does not erase the CP exit block itself. CP will not erase any CP exit blocks until the next IPL.
6. For more information about associating and disassociating entry points and external symbols, see [z/VM: CP Exit Customization](#).

## Responses

### Response 1:

To empty the list of entry point names and external symbols for exit point X'7DF', enter the following series of commands:

```
query exit 7df
Exit Status      Calls      Returns Seconds
07DF Enabled      8          8 10.008833
      EPName      Attempts   Calls Seconds
      HCPsrc00    8          8 0.002598
      HCPsrcZZ    8          0 0.000000
      HCPsrc08    8          4 0.002833
      HCPsrc99    6          6 10.002668
Ready;

disassociate exit 7df
Ready;

query exit 7df
Ready;
Exit Status      Calls      Returns Seconds
07DF Enabled      8          8 10.008833
```

### Response 2:

To empty the list of entry point names for the uppercase messages belonging to component AC\$, enter the following series of commands:

```
query cplanglist associated
Language: AMENG Component: AC$ EPName: AC$CSCT1
Language: UCENG Component: AC$ EPName: AC$CSCT2
Ready;

disassociate msgs component ac$ language uceng
Ready;

query cplanglist associated
Language: AMENG Component: AC$ EPName: AC$CSCT1
Ready;
```

### Response 3:

To empty the list of entry point names for all the messages belonging to component AC\$, enter the following series of commands:

```
query cplanglist associated
Language: AMENG Component: AC$ EPName: AC$CSCT1
Language: UCENG Component: AC$ EPName: AC$CSCT2
Ready;

disassociate msgs component ac$ language *eng
Ready;

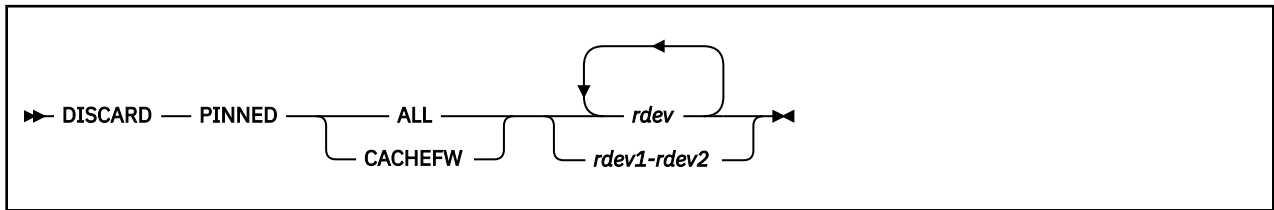
query cplanglist associated
```

There are no associated message repositories  
Ready;

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP1014E A required option is missing - {COMPONENT | LANGUAGE | COMPONENT, LANGUAGE}
- HCP2752E Exit number *exit* does not exist
- HCP2752E Exit numbers do not exist
- HCP2752E Exit numbers *exit1-exit2* do not exist
- HCP2754E Language *xxxxx* for Component *yyy* does not exist
- HCP2778E {Language|Component} identifier *id* is not valid
- HCP6704E Missing token at end of line
- HCP6706E Invalid {string|CP Exit number} - {*string*|*exit*}

## DISCARD PINNED



### Authorization

Privilege Class: B

### Purpose

Use DISCARD PINNED to discard pinned data that is stored in a subsystem cache or nonvolatile storage (NVS). This command is valid for cache storage subsystems that support the cache fast write or DASD fast write functions.

### Operands

#### ALL

indicates that all pinned data for the specified devices is to be discarded.

#### CACHEFW

indicates that only pinned cache fast write data for the specified devices is to be discarded.


**Note:** For IBM DASD subsystems, this operand is being allowed for compatibility. Using the CACHEFW operand will have the same effect as the ALL operand. All pinned data will be discarded.

#### *rdev*

#### *rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers of devices for which data is to be discarded.

### Usage Notes

1. The DISCARD PINNED command is only valid for cache storage subsystems that support the cache fast write or DASD fast write functions. An error message is generated if the specified subsystem does not support any of these functions.
2. This command is used as a recovery tool to remove pinned data from cache and nonvolatile storage (NVS).
3.  **Attention:** This command results in a loss of data. Incorrect use of this command can result in unintended loss of data.
4. This command may be issued to offline devices.

### Responses

Response 1:

The following response is displayed for each specified device when you enter DISCARD PINNED ALL:

```
rdev Pinned data has been discarded.
```

If the originator of the command is not the system operator, the following response is issued to the system operator. This response is also repeated for each specified device.

## DISCARD PINNED

```
rdev Pinned data was discarded by userid.
```

### **rdev**

is the real device number of a device for which pinned data has been discarded.

### **userid**

is the ID of the user who issued the command.

Response 2:

The following response is displayed for each specified device when you enter DISCARD PINNED CACHEFW:

```
rdev Pinned cache fast write data has been discarded.
```

If the originator of the command is not the system operator, the following response is issued to the system operator. This response is also repeated for each specified device.

```
rdev Pinned cache fast write data was discarded by userid.
```

### **rdev**

is the real device number of a device for which pinned cache fast write data has been discarded.

### **userid**

is the ID of the user who issued the command.

## Messages

- HCP006E Invalid device type - {rdev/vdev/ldev}
- HCP009E Invalid range - range
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP046E type rdev offline
- HCP319E Error processing subsystem command - cmd for DASD rdev
- HCP332E Invalid control unit type - rdev
- HCP1109E The command command cannot be processed for device rdev because the device is the secondary in a duplex pair.



## DISCONNECT



### Authorization

Privilege Class: Any

### Purpose

Use DISCONNECT to disconnect your display from the host system without stopping operations in your virtual machine.

### Operands

#### HOLD

causes the non-SNA TTY display terminal telecommunication connection to remain in effect after your session is disconnected. If you specify the HOLD option from a logical device, you disconnect from the logical device without losing the connection between the logical device and the system.

### Usage Notes

1. Once you have issued DISCONNECT, you remain disconnected until you enter the LOGON command. If you have a functional secondary user, that user receives your messages and can enter commands and responses for you. If your virtual machine goes into a disabled wait state, or if a read is posted from the virtual machine or from CP, and you do not have a functional secondary user, you have an installation-defined grace period (usually 15 minutes) in which to log on, or your virtual machine is automatically logged off. If you do not have a functional secondary user, and your virtual machine is in disconnect mode, all “writes” or output messages sent to your virtual console are ignored unless you have issued the CP command SPOOL CONSOLE START.
2. If your connection is broken because of console errors, and you do not have a functional secondary user, CP places your virtual machine in disconnect mode for a grace period defined by the DISCONNECT\_TIMEOUT setting in the SYSTEM CONFIG file. Your virtual machine does not continue to run and you must log on within the grace period if you want to continue operations on your virtual machine and prevent the virtual machine from being FORCED.
3. An error message, without a message ID, is issued when the system disconnects a terminal because of an I/O error.

For example:

```
11:19:20 GRAF 0AB3 DISCONNECT ETC18566 USERS = 59 FORCED BY SYSTEM
```

4. When your virtual machine is reconnected using the usual logon procedure, it is placed in CP console function mode. To resume execution on your virtual machine, enter the BEGIN command.
5. If you specify the HOLD option on an SNA/CCS terminal, the command is not valid and an error message is issued. If you specify the HOLD option on a non-SNA TTY terminal, or a logical terminal, it is accepted and processed. The HOLD option is accepted but ignored on all other terminals.
6. If your installation has coded the E3270HLD feature in the system real I/O configuration file for a TTY display terminal connected to your DACU control unit 7171, the telecommunication connection always remains, regardless of the HOLD option.
7. This command, regardless of the SET RUN setting, causes your virtual machine to run as though BEGIN were entered immediately after DISCONNECT was entered.

## Responses

Response 1:

```
DISCONNECT AT hh:mm:ss zone weekday mm/dd/yy
```

is returned to the user who issued the command.

### hh:mm:ss

identifies the current time.

### zone

identifies the time zone.

### weekday

identifies the current day of the week.

### mm/dd/yy

identifies the current date in month/day-of-month/year format.

The system additionally responds with one of the following, depending on the type of terminal being used:

Type of Terminal	Message
3270 family	Press the ENTER or CLEAR key to continue
SNA/CCS terminal	Press the ENTER key to continue

Pressing the appropriate key on your terminal causes the logo to appear on the terminal, which readies the terminal for the next session. For terminals not listed above, control automatically returns to CP, and the logo is redisplayed.

Response 2:

```
{type {rdev}} DISCONNECT userid USERS = nnnn
{
  {ldev}
  {luname}
  {SYSC}
}
```

is returned to the primary system operator.

### type

identifies the user's display type.

### rdev

### ldev

identifies the real or logical device number associated with the display.

### luname

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

### SYSC

identifies the system console.

### userid

identifies the user who has disconnected from the system.

### USERS = nnnnn

identifies the total number of users remaining on the system.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP059E {XAUTOLOG|AUTOLOG} failed for *userid* - {IPL missing|IPL failed|accounting errors|incorrect password|not authorized}
- HCP197I Secondary user is not available

# DISPLAY

---

## Purpose

Use the DISPLAY commands to see data from the following machine components, both real and virtual:

- The contents of first-level (host) storage
- The contents of second-level storage (the storage that appears real to the operating system running in your virtual machine)
- The contents of third-level storage (the storage that appears virtual to the operating system running in your virtual machine)
- The old and new PSWs, and interrupt information for the following interrupts:
  - SVC
  - I/O
  - PROGRAM
  - EXTERNAL
  - MACHINE CHECK
- Storage keys
- Linkage stack
- The contents of these registers:
  - Access
  - Control
  - Floating-point
  - Floating-point control
  - General purpose
  - Prefix
- Subchannel information blocks (SCHIBS)
- The current program status word (PSW)

## Usage Notes

(General for DISPLAY)

1. Using a single command, you can display the contents of multiple storage locations, registers, and control words.

Follow these rules to execute the command properly:

- a. Do not put blanks between operands that relate to a single DISPLAY request, except for PSW and SCHIB, which require at least one blank between the first and second operands. For example, if you want to display hexadecimal location 40, enter **display H40**.
  - b. Separate each set of display requests from others by at least one blank. For example, if you want to display hexadecimal location 40 and the contents of register 5, enter **display H40 G5**.
2. The defaults for the following DISPLAY operands are:

Operand	Default Before Rounding
G	0-end
G-	0-end

## DISPLAY

Operand	Default Before Rounding
G-5	0-G5
G5-	G5-end
G5	G5
T	T0
T-	T0-end
T-1000	T0-1000
T1000-	T1000-end
T1000	T1000
PSW	Current

3. When displaying a range of addresses or subchannels, the command operand must be followed by an address, a range, or by a hyphen (-), colon (:), or period (.). When displaying registers, the command operand can stand alone.
4. If you enter a command line that contains an operand that is not valid, the DISPLAY command terminates when it encounters that operand. Any previous valid operands are processed before termination occurs.
5. The DISPLAY command lets you display storage in the following formats:
  - Hexadecimal string/word segments
  - EBCDIC or ASCII translation
  - Hexadecimal word segments with EBCDIC translation
  - Instruction
  - Storage key

**Note:** The EBCDIC translation is displayed only when the storage location contains the hexadecimal representation contained in a system translation table. This table is tailorable by the system programmer via the system configuration file. The IBM supplied table supports translations for the hexadecimal representation of the digits 0 through 9, the uppercase and lowercase letters A through Z, and a number of special characters. In all other cases, a period (.) is displayed.

If you enter multiple operands on a single command line to display storage and do not specify the format that you want the DISPLAY command processor to use, the last format type you explicitly enter on the command line defines the type of display you receive. That is, whenever you explicitly enter a display type, you implicitly define the default for subsequent operands of the current display function. If you do not enter any display format type on the command line, the system defaults are employed.

For example, the command **display 10 20 T40 80 G12 5 L60-100** displays the following, respectively:

- Hexadecimal location 10 without EBCDIC translation
  - Hexadecimal location 20 without EBCDIC translation
  - Hexadecimal location 40 with EBCDIC translation
  - Hexadecimal location 80 with EBCDIC translation
  - The contents of general register 12
  - The contents of general register 5
  - Hexadecimal locations 60 through 100 without EBCDIC translation.
6. Use the DISPLAY command to display the contents of various storage locations, registers, and control words. You can examine this information during program execution, and may be able to determine the cause of program errors.

To do this, you would typically define a TRACE trap, using the TRACE command, that will stop program execution at a specified point. Once program execution suspends, your virtual machine enters the CP

environment, and you can enter the DISPLAY command. Alternatively, you can process the DISPLAY command directly from TRACE using the TRACE command CMD option. For more information, see “TRACE: Options” on page 1949.

- The DISPLAY storage command supports indirect addressing. Indirect addressing tells the command to use the **contents** of an address you specified instead of the address itself. This can help reduce the number of displays you issue and the tedious tracking of address pointers. You specify indirect addressing with an **&** (for 64-bit) or a **%** (for 31-bit).

The following is an example of double indirection using the % for 31-bit addressing. First, the storage is shown using the usual DISPLAY without indirect addressing. Following that is the same storage displayed using indirect addressing with only one DISPLAY command (and without having to know that the information you want is at address 3F1AB40). The information displayed is a LNGBK control block for the system default language using a pointer from the CP nucleus SYSCM area.

```
D 3EF1030
R03EF1030 0000524C

D 524C
R0000524C 03F1AB40

D T3F1AB40.80
```

```
R03F1AB40 00000000 00000000 00000000 00000000 06 *.....*
R03F1AB50 to 03F1AB5F suppressed line(s) same as above ...
R03F1AB60 00390000 00000000 C1D4C5D5 C7000000 *.....AMENG...*
R03F1AB70 00000000 03F1A878 00000000 0041E000 *.....1y.....*
R03F1AB80 C8C3D7D4 C5E24040 00000000 00000000 *HCPMES .....*
R03F1AB90 00000000 00000000 00000000 00000000 *.....*
R03F1ABA0 to 03F1ABAF suppressed line(s) same as above ...
R03F1ABB0 4CD3D5C7 4C4C4C4C 80000070 00D5D3E3 *<LNG<<<<.....NLT*
```

```
D T3EF1030%%.80
```

```
R03F1AB40 00000000 00000000 00000000 00000000 06 *.....*
R03F1AB50 to 03F1AB5F suppressed line(s) same as above ...
R03F1AB60 00390000 00000000 C1D4C5D5 C7000000 *.....AMENG...*
R03F1AB70 00000000 03F1A878 00000000 0041E000 *.....1y.....*
R03F1AB80 C8C3D7D4 C5E24040 00000000 00000000 *HCPMES .....*
R03F1AB90 00000000 00000000 00000000 00000000 *.....*
R03F1ABA0 to 03F1ABAF suppressed line(s) same as above ...
R03F1ABB0 4CD3D5C7 4C4C4C4C 80000070 00D5D3E3 *<LNG<<<<.....NLT*
```

- If the DISPLAY command processor is suppressing identical lines, it issues a response every time a segment boundary is crossed, rather than waiting for a nonidentical line.
- If you issue DISPLAY with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the DISPLAY is rejected and error message HCPCDL6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CLeAr option
  - SET MACHine (to a different machine architecture)

## Messages

(General for DISPLAY)

## DISPLAY

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP005E Invalid hexdata - *hexdata*
- HCP009E Invalid range - *range*
- HCP010E Invalid register - *register*
- HCP026E Operand missing or invalid
- HCP160E Hexloc *hexloc* exceeds storage
- HCP164E Hexloc *hexloc* non-addressable storage
- HCP260E {Command|*command* command} processing cannot complete due to errors.
- HCP267E A subchannel number was not supplied or is invalid.
- HCP410E CP ENTERED; PAGING ERROR
- HCP422E The same option was specified twice.
- HCP1001E An operand is missing for *option*.
- HCP1003E {*CP module name*|*rdev*|*ldev*} {is|was} not found.
- HCP1459E The virtual machine is placed in check-stop or system check-stop state due to a system malfunction with CPU *nn*.
- HCP6150E An option has been specified that is not valid in {ESA/390|ESA/XC|z/Architecture|z/XC} mode.
- HCP6151E Storage {operand} is not addressable because of { a protection | an addressing | a segment translation | a page translation | a translation specification | an ASN translation | an AFX translation | an ASX translation | an ALET translation | an ALEN translation | an ALE sequence | an ASTE validity | an ASTE sequence } exception.
- HCP6155E Storage is not addressable because of an invalid PSW.
- HCP6156E {ASIT *asit*|Address space identifier *spaceid*} does not designate an address space to which you have the appropriate access.
- HCP6525E Authorization request failed.
- HCP6578E Invalid {access register|ASIT|address space identifier} - *option*

# DISPLAY ACCESS LIST

►► Display — ACCList ◄◄

## Authorization

Privilege Class: G

## Purpose

Use the DISPLAY ACCESS LIST command to display entries in your virtual machine's host access list.

## Usage Notes

1. If you write a program or EXEC that parses the output of this command, you should avoid a dependency on the ordering of the columns in the output. Your program or EXEC should use the heading line to determine which column contains a particular piece of information.

## Responses

Response 1:

```

ALET      ACC  FAULTS  ASIT      SPACE IDENTIFICATION
ttttttt  {R/O} {ASYNCH} {aaaaaaaaaaaaaaaa} sssssssssssssssssssssssssssssss
          {R/W} {SYNCH } {**REVOKED** }
nnnn TOTAL ENTRIES:  vvvv VALID, rrrr REVOKED, uuuu UNUSED

```

### **ttttttt**

is the access list entry token (ALET) corresponding to the access list entry (8 hexadecimal digits).

### **R/O**

indicates that the access list entry can only be used for read references to the address space.

### **R/W**

indicates that the access list entry can be used for read and write references to the address space.

### **ASYNCH**

indicates that page-fault handshaking applies to the access list entry.

### **SYNCH**

indicates that page-fault handshaking does not apply to the access list entry.

### **aaa...aaa**

is the address space identification token (ASIT) for the address space designated by the access list entry (16 hexadecimal digits).

### **\*\*REVOKED\*\***

indicates that the access list entry previously designated an address space, but access to that address space has been revoked (the access list entry is in the revoked state). The SPACE IDENTIFICATION field is left empty for revoked access list entries.

### **sss...sss**

is the space ID for the address space designated by the access list entry (3 to 33 characters).

### **nnnn TOTAL ENTRIES**

is the total number of entries in the host access list.

### **vvvv VALID**

is the number of entries in the host access list that designate address spaces, and can be used to address data in those address spaces (access list entries that are in the valid state).

## DISPLAY ACCESS LIST

### ***rrrr* REVOKED**

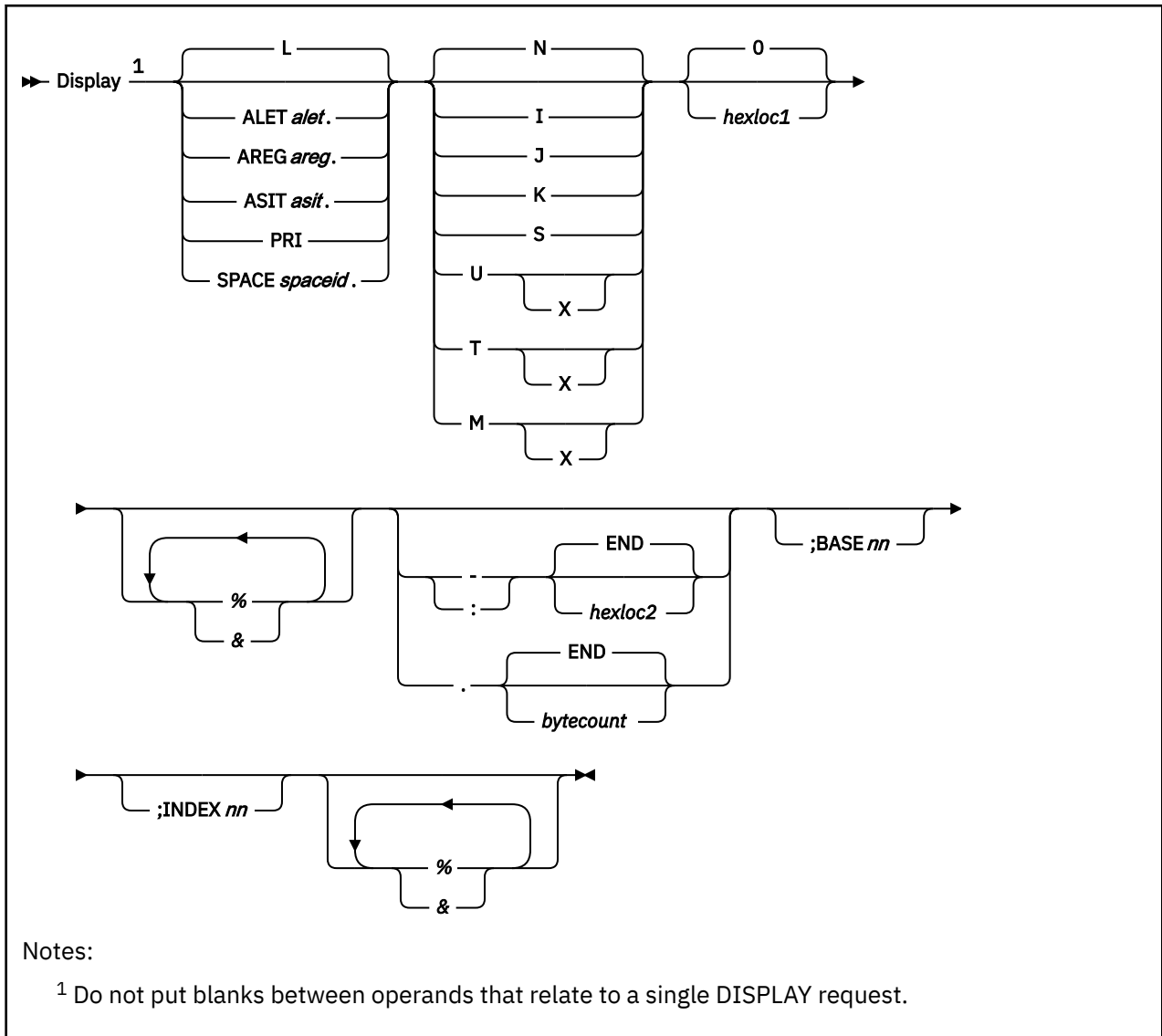
is the number of entries in the host access list that previously designated address spaces, but which can no longer be used to address data because the program's privileges to do so have been withdrawn (access list entries that are in the revoked state).

### ***uuuu* UNUSED**

is the number of entries in the host access list that remain available for use (access list entries that are in the unused state).



## DISPLAY (Guest Storage - ESA/XC, z/XC)



### Authorization

Privilege Class: G

### Purpose

Use DISPLAY (Guest Storage - ESA/XC, z/XC) to display the storage contents of an ESA/XC or z/XC virtual machine.

### Operands

**L**

displays the contents of storage based on the setting of the address-space-control field in the guest PSW. This is the default.

If the virtual CPU is in the primary-space mode, the contents of real storage in the host-primary address space are displayed. If the virtual CPU is in the access register mode and a nonzero BASEnn

operand is also specified, the contents of real storage in an access-register-specified address space are displayed.

If the virtual CPU is in the access register mode, but either the *BASEnn* operand is not specified or is specified as 'BASE0', the contents of real storage in the host-primary address space are displayed.

**ALETalet.**

displays the contents of real storage in an access list entry token (ALET) specified address space. The *alet* may be 1 to 8 hexadecimal digits; leading zeros are optional. The ALET is used in conjunction with the virtual machine's host access list to determine the address space to display.

**AREGareg.**

displays the contents of real storage in an access register specified address space. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The ALET within the access register is used in conjunction with the virtual machine's host access list to determine the address space to display.

**ASITasit.**

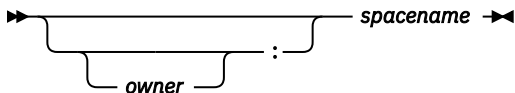
displays the contents of absolute storage in the address space identified by the specified address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

**PRI**

displays the contents of real storage in the host-primary address space. This is the address space from which the virtual CPU is fetching instructions at the time the command is processed.

**SPACEspaceid.**

displays the contents of absolute storage in the address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the address space, and *spacename* is the 1-character to 24-character address space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

**N**

displays the contents of storage in hexadecimal without character translation. This is the default. Storage is displayed in word segments. An address range is displayed starting at the fullword in which the specified address resides. Storage is displayed in fullword increments.

The storage key is displayed on the first line, as well as at every page boundary.

**I**

displays the contents of storage in instruction format. The specified address is rounded down to the nearest halfword boundary.

The storage key is displayed on the first line, as well as at every page boundary.

**J**

stores unformatted storage contents in a response buffer. An address range is stored starting at the byte location specified. If no length or end address is specified, four bytes are stored in the response buffer by default.

When the response buffer is smaller than the response length, a zero-length response is returned, with condition code one to indicate insufficient buffer length.

The operand is accepted only when the DISPLAY command is invoked via DIAGNOSE X'08' with the response-to-buffer option specified. Unlike other invocations of DIAGNOSE X'08', this form of the command does not store a trailing line-end (x'15') character. See usage note ["13"](#) on page 342

for an example of this invocation. See *z/VM: CP Programming Services* for more information about DIAGNOSE X'08'.

**K**

displays the storage keys in hexadecimal, the block-usage state, and the block-content state for all page frames or partial page frames included in the specified address range.

**S**

displays the contents of storage in hexadecimal translation. Storage is displayed as a string starting at the location specified. The specified address range is not boundary-aligned.

The storage key is displayed on the first line, as well as at every page boundary.

**T**

displays the contents of storage in hexadecimal and character form. Storage is displayed in word segments. An address range is displayed starting at the X'10'-byte boundary in which the specified address resides. X'10' bytes of storage are displayed per line. The default display is in EBCDIC; the display is in ASCII if the X option is specified.

The storage key is displayed on the first line, as well as at every page boundary.

**U**

displays the contents of storage in character form. Storage is displayed starting at the address specified as an EBCDIC string, or as an ASCII string if the X option is specified. The specified address range is not boundary-aligned.

The storage key is displayed on the first line, as well as at every page boundary.

**M**

displays the contents of storage in hexadecimal and character form. A header is displayed as the first line, and at every page boundary and contains the high-order 32 bits of the 64-bit address along with the storage key. Storage is displayed in word segments. An address range is displayed starting at the X'10' byte boundary in which the specified address resides. X'10' bytes of storage are displayed per line. The hexadecimal display is followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

**O**

***hexloc1***

specifies the starting storage location that is to be displayed. The value *hexloc1* may be 1 to 16 hexadecimal digits; leading zeros are optional. The default is zero.

If neither the BASE or INDEX operand is used, *hexloc1* is the address of the first location to be displayed. This address is treated as a 64-bit address, regardless of the addressing mode of the virtual CPU.

If either the BASE or the INDEX operand is used, *hexloc1* is a displacement rather than an address. The first location to be displayed is specified by the sum of *hexloc1* and the contents of the registers specified with the BASE operand or the INDEX operand or both. If general register 0 is specified in the BASE or INDEX operand, it is always treated as containing zero when this sum is being formed. The sum is treated as a 24-bit, 31-bit, or 64-bit address according to the addressing mode of the virtual CPU.

Storage locations and the minimum amount displayed are rounded according to the following conventions (in hexadecimal):

Option	Start	Minimum Length
T	10	10
M	10	10
K	800	800
S	1	1
I	2	2 - 6 depending on the instruction

Option	Start	Minimum Length
J	1	1 (default length is 4)
N	4	4
U	1	1

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the DISPLAY command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the DISPLAY command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**-hexloc2**

**:hexloc2**

**-END**

**:END**

is the last of a range of hexadecimal storage locations whose contents are to be displayed. If specified, *hexloc2* must be equal to or greater than *hexloc1* and must be within the bounds of the address space to be displayed. The address *hexloc2* may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. A colon (:) means the same as a dash (-) when you are specifying a range of addresses.

If neither the BASE or INDEX operand is used, *hexloc2* is the address of the last location to be displayed. This address is treated as a 64-bit address, regardless of the addressing mode of the virtual CPU.

If either the BASE or the INDEX operand is used, *hexloc2* is a displacement rather than an address. The last location to be displayed is specified by the sum of *hexloc2* and the contents of the registers specified with the BASE and INDEX operands. If general register 0 is specified in the BASE or INDEX operand, it is always treated as containing zero when forming this sum. The sum is treated as a 24-bit, 31-bit, or 64-bit address according to the addressing mode of the virtual CPU.

Specify END if you want to display the remaining contents of storage. END is the default if you specify a dash (-) or a colon (:) without an address.

**.bytcoun**

**.END**

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at *hexloc1*) to be displayed. If specified, *bytcoun* must have a value of at least 1 and may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional.

Specify END if you want to display the remaining contents of storage. END is the default if you specify period (.) without a byte count value.

**BASEnn**

displays the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified general register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

If the virtual CPU is in access register mode, and the L operand is specified (or defaulted), *nn* designates an access register/general register pair. Access register *nn* is implicitly used, in the same manner as if the AREG operand had been specified, in conjunction with the virtual machine's host

access list to determine the address space to display. If the virtual CPU is not in the access register mode, or if one of the other space-designation operands (PRI, AREG, ALET, SPACE, ASIT) is specified, access register *nn* is not implicitly used.

### INDEX $nn$

displays the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

### Usage Notes

- Special authorization is required to use this command to view any address space other the host-primary address space of the virtual CPU. In order to use this command to display storage in an address space, one of the following conditions must be satisfied:
  - Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.
  - If the address space is owned by another virtual machine, your virtual machine must have read-only or read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.

Authorization is granted on a space-by-space basis, so that you may be authorized to view one address space of a virtual machine, but may not be authorized to view a different space.
- When running with a saved segment, you can display storage locations outside the range of your virtual machine size if they are within the saved segment. If locations exist between the upper limit of your virtual machine and the address at which the saved segment was saved, any attempt to display those locations (or associated keys) causes the non-addressable storage response to be displayed.
- The operands L, PRI, AREG, and ALET specify the host-access-register translation (ART) process. The ART process performs special-case translation processes for certain ALET values and access registers:
  - The ALET X'00000000' always indicates the host-primary address space.
  - Access register 0 is always treated as containing ALET X'00000000'. The contents of access register 0 are not used during host-access-register translation processing.
- General register 0 is always treated as containing the value 0 when used as the base or index register with the BASE or INDEX operands. The contents of general register 0 are never used.
- In the XC environment, special rules govern the conversion of real addresses into absolute addresses. In some cases, prefixing is performed as usual to convert a real address to an absolute address, while in other cases prefixing is not performed and real addresses are treated unchanged as absolute addresses.

These rules have an effect on the interpretation of the addresses that specify the range of storage locations to be displayed:

- When storage is specified using the ALET or AREG operand and the ALET value used is not X'00000000', prefixing is not performed on the addresses that specify the range of storage in the target address space.
  - When storage in the host-primary address space is specified by using the PRI operand, or by using the ALET or AREG operand when the ALET value used is X'00000000', prefixing is performed as usual on the addresses that specify the range of storage in the host-primary address space.
- Although the R operand is not shown in the syntax for this command, it is accepted for compatibility with the DISPLAY (Guest Storage) command used for XA and ESA virtual machines. The R operand perform the same function as the L operand and displays storage from the host-primary address space.
  - The BASE operand is intended to be used in conjunction with the L operand. It lets you display the storage used by the virtual machine in one less step than would otherwise be possible. Without the BASE operand, you may need to display the contents of a general register, add the displacement,

and enter the DISPLAY command to examine storage the program is using. With the use of access registers, the BASE option additionally lets you examine storage without needing to know whether the program is currently in access register mode. This is done automatically for you by CP.

8. When multiple DISPLAY requests are entered on a single line, the BASE and INDEX operands are not applied to subsequent DISPLAY requests on the command line. They must be explicitly entered for each DISPLAY request.
9. During DISPLAY command processing, key-controlled protection is not used. However, all of the host-related protection mechanisms, such as those applied during host-access-register translation, are used as normal.
10. If the AREG or ALET operand is specified, the DISPLAY command uses the host-access-register translation (ART) process to determine the address space to display. This translation process may fail because of one of several program exception conditions. If the translation does not succeed, the current request is rejected and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the exception condition.
11. When the ASIT or SPACE operand is specified, the DISPLAY request fails if the specified address space does not exist, or if you are not authorized to access the address space. The current request is rejected and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
12. If you use an underscore, it must be followed by 8 hexadecimal digits.
13. The J operand is useful in an adjunct context to retrieve memory from the principal configuration. For example, the following line

```
pgmold=Diag(8,'CPU * DISPLAY RJ28.8')
```

can be included in a REXX exec to capture the z/Architecture program old PSW.

14. For more information, see [“Usage Notes” on page 331](#).

## Responses

### Notes:

1. The value *hexloc* is a hexadecimal storage address.
2. The character displayed for the variable *t* in the string *thexloc* indicates the type of storage:
  - R indicates real storage (second-level storage).
  - V indicates virtual storage (third-level storage).

Response 1:

```
Rhexloc word1 word2 word3 word4 [key] [*TRANSLATION*]
```

is the response you receive for the T or N option. The translation is displayed only if you have also entered the T option and is in either EBCDIC or (if the TX option is specified) ASCII. The storage key is displayed on the first line of a request, and at every page boundary.

Response 2:

```
Rhexlochi_ Storage Key - [key]  
hexlocl_ word1 word2 word3 word4 [*TRANSLATION*]
```

is the response you receive for the M option. The translation is displayed in either EBCDIC or (if the MX option is specified) ASCII. The storage key is displayed on the first line, and at every page boundary.

Response 3:

```
Rhexloc *TRANSLATION* [key]
```

is the response you receive for the U option. The value *hexloc* is the location of the first data byte to be displayed. A minimum of 1 character and a maximum of 32 characters are displayed on each line. The contents of storage are displayed in translated form and are in either EBCDIC or (if the UX option is specified) ASCII. Periods in your display represent unprintable characters. The storage key is displayed on the first line of a request and at every page boundary.

Response 4:

```
Rhexloc mnemonic x-x mnemonic x-x mnemonic x-x [key]
```

is the response you receive for the I option. The value *hexloc* is the location of the first instruction to be displayed. The value *mnemonic* is a character representation of the instruction code; this is a 7-character field in which the mnemonic is left-justified. The expression *x-x* is the actual hexadecimal contents of storage and is 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. A minimum of 1 instruction and a maximum of 3 instructions are displayed on each line. The storage key is displayed on the first line of a request and at every page boundary.

Response 5:

```
Rhexloc stringhexadecimaldata [key]
```

is the response you receive for the S option. The storage key is displayed on the first line of a request and at every page boundary.

Response 6:

```
thexloc1 TO hexloc2 KEY = key STATE = uc
```

is the response you receive for the K option. The values *hexloc1* and *hexloc2* are storage locations, *key* is the associated storage key for *hexloc1*, and *uc* is a two-letter code describing the block-usage and block-content state:

**u**

is a code for the block-usage state:

**Code**

**Meaning**

**S**

Stable

**U**

Unused

**V**

Volatile

**P**

Potentially volatile

**c**

is a code for the block-content state:

**Code**

**Meaning**

**R**

Resident

**P**

Preserved

**Z**

Logically zero

For more information about block-usage and block-content states, see [Collaborative memory management assist](#) in *z/VM: CP Programming Services*.

## DISPLAY (Guest Storage - ESA/XC)

A response is sent for every contiguous address range that has unique storage keys and whenever a segment boundary is crossed.

Response 7:

The PSW is used to resolve a logical address. An invalid PSW causes the current request to be rejected and processing to continue with the next request (if any) on the command line. In all cases, a reset PSW (all-zeros) is accepted. The rejected operands are displayed along with a response indicating the PSW is invalid. The response is:

```
addr is not addressable because of an invalid PSW.
```

where *addr* is the rejected operand or operands; it specifies an address space and a storage location within that address space.

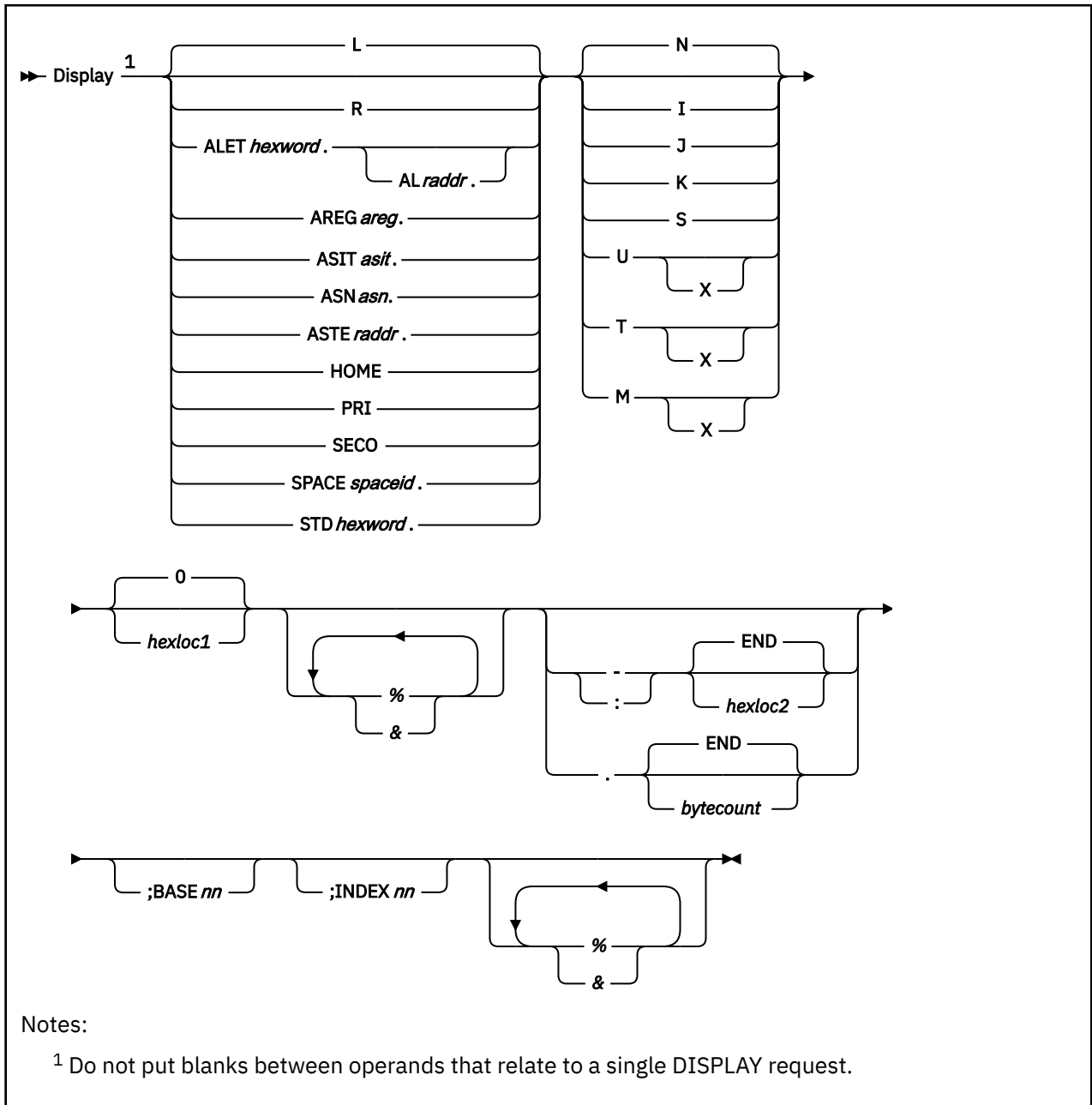
Response 8:

If your display request includes a request to display identical lines, you receive the following response:

```
Rhexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...
```



## DISPLAY (Guest Storage - ESA/390)



### Authorization

Privilege Class: G

### Purpose

Use DISPLAY (Guest Storage - ESA/390) to display the storage contents of an ESA/390 virtual machine.

## Operands

### L

displays the contents of second-level storage (storage that appears real to your virtual machine) if the virtual machine is operating in DAT OFF mode, or displays the contents of third-level storage (storage that appears virtual to your virtual machine) if the virtual machine is operating in DAT ON mode. This is the default.

The address space selected for display of third-level storage corresponds with the setting of the address space control field in the guest PSW and is the primary, secondary, or (if the virtual machine is an ESA virtual machine) home space. When the PSW indicates access register mode, the primary space is selected for display unless a nonzero BASExx operand is also specified and the virtual machine is an ESA virtual machine, in which case an access register specified address space is displayed.

### R

displays the contents of second-level storage (storage that appears real to your virtual machine).

#### **ALE***hexword*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified access list entry token (ALET). The ALET is used in conjunction with guest access register translation tables to locate the guest address space to display.

#### **AL***raddr*.

specifies an access list (AL) to be used in conjunction with the ALET to locate the guest address space. The guest real address of the access list may be 1 to 8 hexadecimal digits.

#### **AREG***areg*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the contents of the specified access register. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The ALET within the access register is used in conjunction with guest access register translation tables to locate the guest address space to display.

#### **ASIT***asit*.

displays the contents of absolute storage in the CP-created address space identified by the specified address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

#### **ASN***asn*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified address space number (ASN). The *asn* may be 1 to 4 hexadecimal digits. It is used in conjunction with guest ASN-translation tables to locate a guest address space to display.

#### **ASTE***raddr*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the segment table designation contained within the specified ASN second table entry (ASTE) located at guest real address *raddr*.

#### **HOME**

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 13.

#### **PRI**

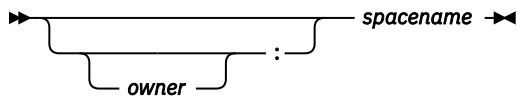
displays the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 1.

#### **SECO**

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 7.

#### **SPACE***spaceid*.

displays the contents of absolute storage in the CP-created address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the CP-created address space, and *spacename* is the 1-character to 24-character address space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

**STDhexword.**

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified guest segment table designation (STD).

**N**

displays the contents of storage in hexadecimal without EBCDIC translation. This is the default. Storage is displayed in word segments. An address range is displayed starting at the fullword in which the specified address resides. Storage is displayed in fullword increments.

The storage key is displayed on the first line, as well as at every page boundary.

**I**

displays the contents of storage in instruction format. The specified address is rounded down to the nearest halfword boundary.

The storage key is displayed on the first line, as well as at every page boundary.

**J**

stores unformatted storage contents in a response buffer. An address range is stored starting at the byte location specified. If no length or end address is specified, four bytes are stored in the response buffer by default.

When the response buffer is smaller than the response length, a zero-length response is returned, with condition code one to indicate insufficient buffer length.

The operand is accepted only when the DISPLAY command is invoked via DIAGNOSE X'08' with the response-to-buffer option specified. Unlike other invocations of DIAGNOSE X'08', this form of the command does not store a trailing line-end (x'15') character. See usage note "15" on page 350 for an example of this invocation. See [z/VM: CP Programming Services](#) for more information about DIAGNOSE X'08'.

**K**

displays the storage keys in hexadecimal, the block-usage state, and the block-content state for all page frames or partial page frames included in the specified address range.

**S**

displays the contents of storage in hexadecimal translation. Storage is displayed as a string starting at the location specified. The specified address range is not boundary-aligned.

The storage key is displayed on the first line, as well as at every page boundary.

**U**

displays the contents of storage in character form. Storage is displayed starting at the address specified as an EBCDIC string or as an ASCII string if the X option is specified. The specified address range is not boundary-aligned.

The storage key is displayed on the first line, as well as at every page boundary.

**T**

displays the contents of storage in hexadecimal and character form. Storage is displayed in word segments. An address range is displayed starting at the X'10'-byte boundary in which the specified address resides. X'10' bytes of storage are displayed per line. The default display is in EBCDIC; the display is in ASCII if the X option is specified.

The storage key is displayed on the first line, as well as at every page boundary.

**M**

displays the contents of storage in hexadecimal and character form. A header is displayed as the first line, and at every page boundary, which contains the high-order 32 bits of the 64-bit address along with the storage key. Storage is displayed in word segments. An address range is displayed starting at the X'10' byte boundary in which the specified address resides. X'10' bytes of storage are displayed per line. The hexadecimal display is followed by the storage as an EBCDIC string or as an ASCII string if the X option is specified.

**0**

**hexloc1**

is the starting storage location that is to be displayed. The address *hexloc1* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability, leading zeros are optional. The default is zero.

Storage locations and the minimum amount displayed are rounded according to the following conventions (in hexadecimal):

Option	Start	Minimum Length
T	10	10
M	10	10
K	800	800
S	1	1
I	2	2 - 6 depending on the instruction
J	1	1 (default length is 4)
N	4	4
U	1	1

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the DISPLAY command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the DISPLAY command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**-hexloc2**

**:hexloc2**

**-END**

**:END**

is the last of a range of hexadecimal storage locations whose contents are to be displayed. If specified, *hexloc2* must be equal to or greater than *hexloc1* and must be within the virtual machine storage size. The address *hexloc2* may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. A colon (:) means the same as a dash (-) when you are specifying a range of addresses.

Specify END if you want to display the remaining contents of storage. END is the default if you specify a dash (-) or a colon (:) without an address.

**.bytecount****.END**

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at *hexloc1*) to be displayed. If specified, the *bytecount* must have a value of at least 1 and may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional.

Specify END if you want to display the remaining contents of storage. END is the default if you specify a period (.) without a byte count value.

**BASEnn**

displays the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified general register (*nn*). The general register may be specified as a decimal number from 0 to 15 or as a hexadecimal number from 0 to F. *nn* specifies an AR/GR pair if all of the following are true:

- The L operand is specified.
- The virtual machine is an ESA virtual machine in ESA/390 mode (the default mode).
- The virtual CPU is in access register mode.

The specified access register is used, in the same manner as if the *AREGareg.* operand had been specified, in conjunction with guest access-register-translation tables to locate the guest address space to display.

**INDEXnn**

displays the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified register (*nn*). This register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

**Usage Notes**

1. When running with a saved segment, you can display storage locations outside the range of your virtual machine size if they are within the saved segment. If locations exist between the upper limit of your virtual machine and the address at which the saved segment was saved, any attempt to display those locations (or associated keys) causes the non-addressable storage response to be displayed.

Refer to [“Usage Notes” on page 331](#).

2. The operands HOME, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* are valid only if the virtual machine is an XA or ESA virtual machine.
3. When you use the operand *ALETheyword.ALraddr.*, the primary list bit in the ALET is ignored.
4. When the operands L, PRI, SECO, HOME, *ASTERaddr.*, *STDhexword.*, *ASNasn.*, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* are specified, the access-register-translation (ART), the ASN translation, and the dynamic-address-translation (DAT) processes may be used. These contain special-case translation processes, including:
  - ALET 0 always indicates the current primary address space designated by the STD in guest control register 1, regardless of the current contents of the dispatchable-unit access list.
  - ALET 1 always indicates the current secondary address space designated by the STD in guest control register 7, regardless of the current contents of the dispatchable-unit access list.
  - Access register 0 is always treated as containing ALET 0. The contents of access register 0 are not used during access-register-translation processing.
  - General register 0 is always treated as containing the value 0 when used as the base or index register. The contents of general register 0 are never used.
5. The BASE operand is primarily intended to be used in conjunction with the L operand. It lets you display the storage used by the virtual machine in one less step than would otherwise be possible. Without the BASE operand, you may need to display the contents of a general register, add the displacement, and then enter the DISPLAY command to examine storage the program is using. With the use of access registers, the BASE option also lets you examine storage without needing to know whether the program is currently in access register mode. This is done automatically for you by CP.

6. If the *BASEnn* and *INDEXnn* operands are used together, the specified starting address for display is used as a displacement from the address formed from the sum of the contents of the two specified registers.

The values in the registers specified by the *BASEnn* and *INDEXnn* operands are treated as 24-bit or 31-bit real addresses according to the addressing mode (A) bit in the virtual machine's PSW. That is, if the A bit is 1, the value is a 31-bit address; if the A bit is 0, the value is a 24-bit address.

7. During DISPLAY command processing, the authorization mechanisms used during normal address translation are not used.
8. Special authorization is required to use the SPACE or ASIT operand of this command to view any address space other the host-primary address space of the virtual CPU. In order to use this command to display storage in an address space, one of the following conditions must be satisfied:
  - Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.
  - If the address space is owned by another virtual machine, your virtual machine must have read-only or read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.

Authorization is granted on a space-by-space basis, so that you may be authorized to view one address space of a virtual machine, but not to view a different space.

9. The operands V and W used in displaying third-level storage from guest primary and secondary are supported for compatibility. The PRI and SECO operands replace the V and W operands.
10. When multiple DISPLAY requests are entered on a single line, the *BASEnn* and *INDEXnn* operands are not applied to subsequent DISPLAY requests on the command line. They must be explicitly entered for each DISPLAY request.
11. When the ASIT or SPACE operand is specified, the DISPLAY request will fail if the specified address space does not exist, or if you are not authorized to access the address space. The current request is rejected and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
12. The DISPLAY command requires a segment table designation (STD) to display third-level storage (storage that appears virtual to your virtual machine). It may receive the STD as an operand, find it in a control register, or use the access-register translation (ART) process or the address space number (ASN) translation process to obtain the STD. The access-register translation and ASN translation processes may fail because of one of several program exception conditions. If the translation does not succeed, the current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a response indicating the nature of the exception condition.
13. If you use an underscore, it must be followed by 8 hexadecimal digits.
14. If you issue DISPLAY with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the DISPLAY command is rejected and error message HCPCDL6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CLear option
  - SET MACHine (to a different machine architecture)
15. The J operand is useful in an adjunct context to retrieve memory from the principal configuration. For example, the following line

```
pgmold=Diag(8,'CPU * DISPLAY RJ28.8')
```

can be included in a REXX exec to capture the z/Architecture program old PSW.

16. For more information, see [“Usage Notes” on page 331](#).

## Responses

### Notes:

1. The value *hexloc* is a hexadecimal storage address.
2. The character displayed for the variable *t* in the string *thexloc* indicates the type of storage:
  - R indicates real storage (second-level storage).
  - V indicates virtual storage (third-level storage).
3. In responses that include third-level storage, an *Rhexloc* trailer shows the second-level address that corresponds to the third-level address. The trailer is displayed on the first line of a range and at every page boundary.

Response 1:

```
thexloc word1 word2 word3 word4 [key] [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the T or N option. The translation is given only for the T option and is in either EBCDIC or (if the TX option is specified) ASCII. The storage key is displayed on the first line of a request, and at every page boundary.

Response 2:

```
thexlochi Storage Key - [key]  
hexloclo word1 word2 word3 word4 [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the M option. The translation is displayed in either EBCDIC or (if the MX option is specified) ASCII. The storage key is displayed on the first line, and at every page boundary.

Response 3:

```
thexloc *TRANSLATION* [key] [Rhexloc]
```

is the response you receive for the U option. The value *hexloc* is the location of the first data byte to be displayed. A minimum of 1 character and a maximum of 32 characters are displayed on each line. The contents of storage are displayed only in translated form and are in either EBCDIC or (if the UX option is specified) ASCII. Periods in the display represent unprintable characters. The storage key is displayed on the first line of a request and at every page boundary.

Response 4:

```
thexloc mnemonic x-x mnemonic x-x mnemonic x-x [key] [Rhexloc]
```

is the response you receive for the I option. The value *hexloc* is the location of the first instruction to be displayed. The value *mnemonic* is a character representation of the instruction code; this is a 7-character field in which the mnemonic is left-justified. The expression *x-x* is the actual hexadecimal contents of storage and is 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. A minimum of 1 instruction and a maximum of 3 instructions are displayed on each line. The storage key is displayed on the first line of a request and at every page boundary.

Response 5:

```
thexloc stringhexadecimaldata [key] [Rhexloc]
```

is the response you receive for the S option. The storage key is displayed on the first line of a request and at every page boundary.

Response 6:

```
thexloc1 TO hexloc2 KEY = key STATE = uc
```

is the response you receive for the K option. The values *hexloc1* and *hexloc2* are storage locations, *key* is the associated storage key for *hexloc1*, and *uc* is a two-letter code describing the block-usage and block-content state:

**u**

is a code for the block-usage state:

**Code**

**Meaning**

**S**

Stable

**U**

Unused

**V**

Volatile

**P**

Potentially volatile

**c**

is a code for the block-content state:

**Code**

**Meaning**

**R**

Resident

**P**

Preserved

**Z**

Logically zero

For more information about block-usage and block-content states, see [Collaborative memory management assist](#) in *z/VM: CP Programming Services*.

A response is sent for every contiguous address range that has unique storage keys and whenever a segment boundary is crossed.

Response 7:

The PSW is used to resolve a logical address. A PSW that is not valid causes the current request to be rejected and processing to continue with the next request (if any) on the command line. In all cases, a reset PSW (all-zero) is accepted. The rejected operands are displayed along with a response indicating that the PSW is not valid. The response is:

```
addr is not addressable because of a invalid PSW.
```

where *addr* is the rejected operand or operands; it specifies an address space and a storage location within that address space.

Response 8:

The dynamic-address translation (DAT) process uses the segment table designator (STD) to convert a third-level address to a second-level address (storage that appears real to your virtual machine). The translation process may generate one of several exceptions, identifying why one or more pages of storage are not addressable. A range of rejected storage addresses is displayed along with a response indicating the nature of the exception condition. The responses are:

```
thexloc1 to hexloc2 non-addressable storage
thexloc1 to hexloc2 non-addressable storage
- protection exception
thexloc1 to hexloc2 non-addressable storage
- addressing exception
```



```
hexloc1 to hexloc2 non-addressable storage  
- segment translation exception  
hexloc1 to hexloc2 non-addressable storage  
- page translation exception  
hexloc1 to hexloc2 non-addressable storage  
- translation specification exception
```

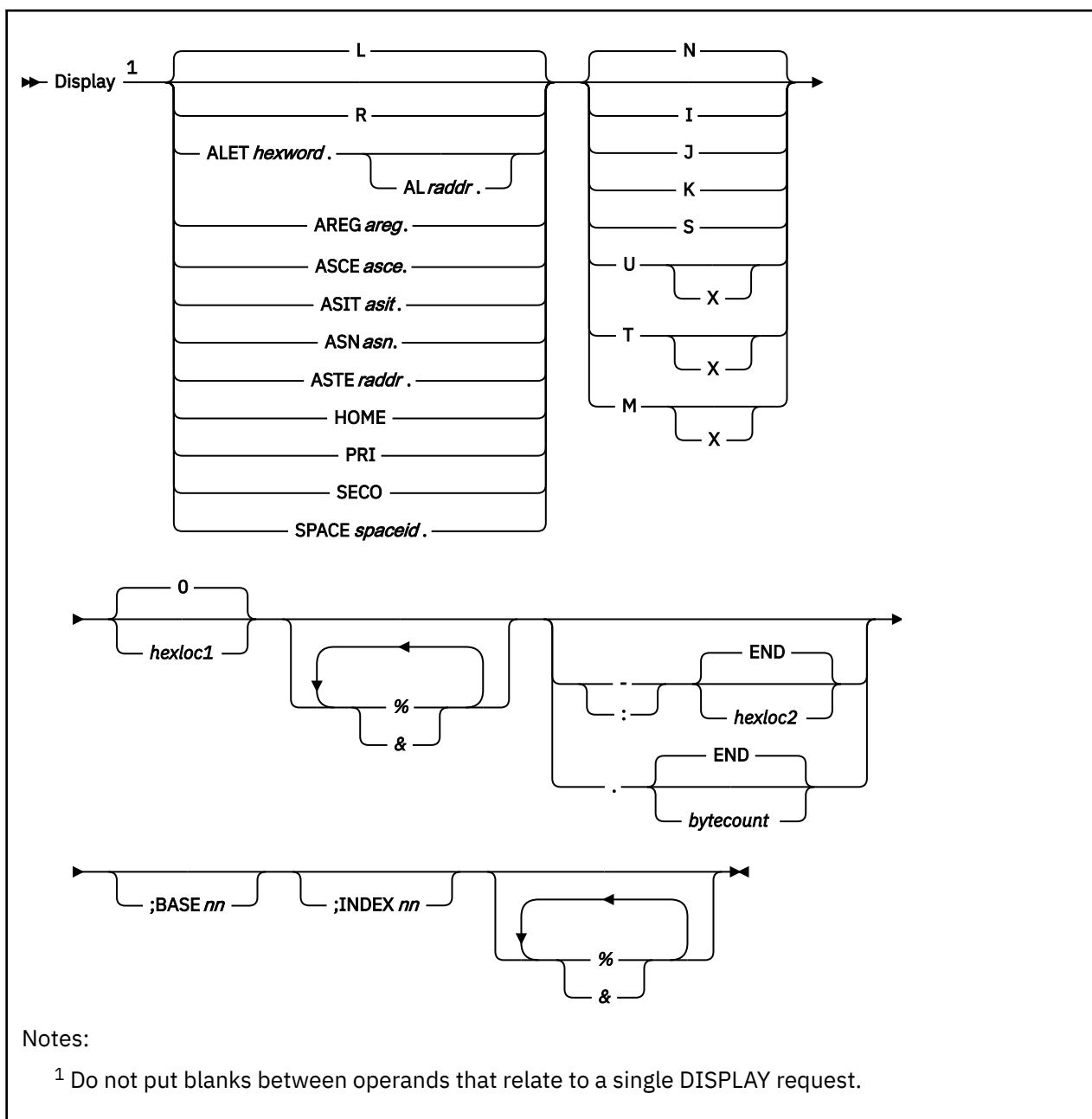
where *hexloc1* is the starting address and *hexloc2* the ending address of the non-addressable storage. The addresses take into account basing, indexing, and wrapping and are padded to 8 digits.

Response 9:

If your display request includes a request to display identical lines, you receive the following response:

```
hexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...
```

## DISPLAY (Guest Storage - z/Architecture)



### Authorization

Privilege Class: G

### Purpose

Use DISPLAY (Guest Storage - z/Architecture) to display the storage contents of a virtual machine in z/Architecture mode.

## Operands

### L

displays the contents of second-level storage (storage that appears real to your virtual machine) if the virtual machine is operating in DAT OFF mode, or displays the contents of third-level storage (storage that appears virtual to your virtual machine) if the virtual machine is operating in DAT ON mode. This is the default.

The address space selected for display of third-level storage corresponds with the setting of the address space control field in the guest PSW and is the primary, secondary, or (if the virtual machine is an ESA virtual machine) home space. When the PSW indicates access register mode, the primary space is selected for display unless a nonzero *BASEnn* operand is also specified and the virtual machine is an ESA virtual machine, in which case an access register specified address space is displayed.

### R

displays the contents of second-level storage (storage that appears real to your virtual machine).

#### **ALET***hexword*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified access list entry token (ALET). The ALET is used in conjunction with guest access register translation tables to locate the guest address space to display.

#### **AL***raddr*.

specifies an access list (AL) to be used in conjunction with the ALET to locate the guest address space. The guest real address of the access list may be 1 to 8 hexadecimal digits.

#### **AREG***areg*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the contents of the specified access register. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The ALET within the access register is used in conjunction with guest access register translation tables to locate the guest address space to display.

#### **ASCE***asce*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified address-space-control element (ASCE). The *asce* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional.

#### **ASIT***asit*.

displays the contents of absolute storage in the CP-created address space identified by the specified address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

#### **ASN***asn*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified address space number (ASN). The *asn* may be 1 to 4 hexadecimal digits. It is used in conjunction with guest ASN-translation tables to locate a guest address space to display.

#### **ASTE***raddr*.

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using the segment table designation contained within the specified ASN second table entry (ASTE) located at guest real address *raddr*.

### HOME

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 13.

### PRI

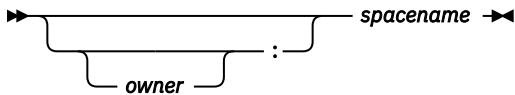
displays the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 1.

### SECO

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 7.

**SPACEspaceid.**

displays the contents of absolute storage in the CP-created address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the CP-created address space, and *spacename* is the 1-character to 24-character address space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

**N**

displays the contents of storage in hexadecimal without character translation. This is the default. Storage is displayed in word segments. An address range is displayed starting at the fullword in which the specified address resides. Storage is displayed in fullword increments.

The storage key is displayed on the first line, as well as at every page boundary.

**I**

displays the contents of storage in instruction format. The specified address is rounded down to the nearest halfword boundary.

The storage key is displayed on the first line, as well as at every page boundary.

**J**

stores unformatted storage contents in a response buffer. An address range is stored starting at the byte location specified. If no length or end address is specified, four bytes are stored in the response buffer by default.

When the response buffer is smaller than the response length, a zero-length response is returned, with condition code one to indicate insufficient buffer length.

The operand is accepted only when the DISPLAY command is invoked via DIAGNOSE X'08' with the response-to-buffer option specified. Unlike other invocations of DIAGNOSE X'08', this form of the command does not store a trailing line-end (x'15') character. See usage note "14" on page 360 for an example of this invocation. See [z/VM: CP Programming Services](#) for more information about DIAGNOSE X'08'.

**K**

displays the storage keys in hexadecimal, the block-usage state, and the block-content state for all page frames or partial page frames included in the specified address range.

**S**

displays the contents of storage in hexadecimal translation. Storage is displayed as a string starting at the location specified. The specified address range is not boundary-aligned.

The storage key is displayed on the first line, as well as at every page boundary.

**U**

displays the contents of storage in character form. Storage is displayed starting at the address specified as an EBCDIC string or as an ASCII string if the X option is specified. The specified address range is not boundary-aligned.

The storage key is displayed on the first line, as well as at every page boundary.

**T**

displays the contents of storage in hexadecimal and character form. Storage is displayed in word segments. An address range is displayed starting at the X'10'-byte boundary in which the specified address resides. X'10' bytes of storage are displayed per line. The default display is in EBCDIC; the display is in ASCII if the X option is specified.

The storage key is displayed on the first line, as well as at every page boundary.

**M**

displays the contents of storage in hexadecimal and character form. A header is displayed as the first line, and at every page boundary, which contains the high-order 32 bits of the 64-bit address along with the storage key. Storage is displayed in word segments. An address range is displayed starting at the X'10' byte boundary in which the specified address resides. X'10' bytes of storage are displayed per line. The hexadecimal display is followed by the storage as an EBCDIC string or as an ASCII string if the X option is specified.

**O**

**hexloc1**

is the starting storage location that is to be displayed. The address *hexloc1* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability, leading zeros are optional. The default is zero.

Storage locations and the minimum amount displayed are rounded according to the following conventions (in hexadecimal):

Option	Start	Minimum Length
T	10	10
M	10	10
K	800	800
S	1	1
I	2	2 - 6 depending on the instruction
J	1	1 (default length is 4)
N	4	4
U	1	1

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the DISPLAY command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the DISPLAY command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**-hexloc2**

**:hexloc2**

**-END**

**:END**

is the last of a range of hexadecimal storage locations whose contents are to be displayed. If specified, *hexloc2* must be equal to or greater than *hexloc1* and must be within the virtual machine storage size. The address *hexloc2* may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. A colon (:) means the same as a dash (-) when you are specifying a range of addresses.

Specify END if you want to display the remaining contents of storage. END is the default if you specify a dash (-) or a colon (:) without an address.

### **.bytecount**

#### **.END**

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at *hexloc1*) to be displayed. If specified, the *bytecount* must have a value of at least 1 and may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional.

Specify END if you want to display the remaining contents of storage. END is the default if you specify a period (.) without a byte count value.

### **BASEnn**

displays the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified general register (*nn*). The general register may be specified as a decimal number from 0 to 15 or as a hexadecimal number from 0 to F. *nn* specifies an AR/GR pair if all of the following are true:

- The L operand is specified.
- The virtual machine is an ESA virtual machine in z/Architecture mode.

**Note:** An ESA virtual machine simulates ESA/390 architecture when initially defined. However, a guest may have the capability to switch the virtual machine from ESA/390 mode to z/Architecture mode.

- The virtual CPU is in access register mode.

The specified access register is used, in the same manner as if the *AREGareg.* operand had been specified, in conjunction with guest access-register-translation tables to locate the guest address space to display.

### **INDEXnn**

displays the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified register (*nn*). This register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

## **Usage Notes**

1. When running with a saved segment, you can display storage locations outside the range of your virtual machine size if they are within the saved segment. If locations exist between the upper limit of your virtual machine and the address at which the saved segment was saved, any attempt to display those locations (or associated keys) causes the non-addressable storage response to be displayed.

Refer to [“Usage Notes” on page 331](#).

2. When you use the operand *ALETheyword.ALraddr.*, the primary list bit in the ALET is ignored.
3. When the operands L, PRI, SECO, HOME, *ASTERaddr.*, *ASCEhexword.*, *ASNasn.*, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* are specified, the access-register-translation (ART), the ASN translation and the dynamic-address-translation (DAT) processes may be used. These contain special-case translation processes, including:
  - ALET 0 always indicates the current primary address space designated by the ASCE in guest control register 1, regardless of the current contents of the dispatchable-unit access list.
  - ALET 1 always indicates the current secondary address space designated by the ASCE in guest control register 7, regardless of the current contents of the dispatchable-unit access list.
  - Access register 0 is always treated as containing ALET 0. The contents of access register 0 are not used during access-register-translation processing.
  - General register 0 is always treated as containing the value 0 when used as the base or index register. The contents of general register 0 are never used.
4. The BASE operand is primarily intended to be used in conjunction with the L operand. It lets you display the storage used by the virtual machine in one less step than would otherwise be possible. Without the BASE operand, you may need to display the contents of a general register, add the displacement, and then enter the DISPLAY command to examine storage the program is using. With

the use of access registers, the BASE option also lets you examine storage without needing to know whether the program is currently in access register mode. This is done automatically for you by CP.

5. If the BASE $nn$  and INDEX $nn$  operands are used together, the specified starting address for display is used as a displacement from the address formed from the sum of the contents of the two specified registers.

The values in the registers specified by the BASE $nn$  and INDEX $nn$  operands are treated as 24-bit, 31-bit, or 64-bit real addresses according to the addressing mode bits (EA and BA - bits 31 and 32) in the virtual machine's PSW. The addressing mode is determined as follows:

Bit 31	Bit 32	Addressing Mode
0	0	24-bit
0	1	31-bit
1	1	64-bit

6. During DISPLAY command processing, the authorization mechanisms used during normal address translation are not used.
7. Special authorization is required to use the SPACE or ASIT operand of this command to view any address space other than the host-primary address space of the virtual CPU. In order to use this command to display storage in an address space, one of the following conditions must be satisfied:
  - Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.
  - If the address space is owned by another virtual machine, your virtual machine must have read-only or read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.

Authorization is granted on a space-by-space basis, so that you may be authorized to view one address space of a virtual machine, but not to view a different space.
8. The operands V and W used in displaying third-level storage from guest primary and secondary are supported for compatibility. The PRI and SECO operands replace the V and W operands.
9. When multiple DISPLAY requests are entered on a single line, the BASE $nn$  and INDEX $nn$  operands are not applied to subsequent DISPLAY requests on the command line. They must be explicitly entered for each DISPLAY request.
10. When the ASIT or SPACE operand is specified, the DISPLAY request will fail if the specified address space does not exist, or if you are not authorized to access the address space. The current request is rejected and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
11. The DISPLAY command requires an address-space-control element (ASCE) to display third-level storage (storage that appears virtual to your virtual machine). It may receive the ASCE as an operand, find it in a control register, or use the access-register translation (ART) process or the address space number (ASN) translation process to obtain the ASCE. The access-register translation and ASN translation processes may fail because of one of several program exception conditions. If the translation does not succeed, the current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a response indicating the nature of the exception condition.
12. If you use an underscore, it must be followed by 8 hexadecimal digits.
13. If you issue DISPLAY with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the DISPLAY command is rejected and error message HCPCDL6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:

## DISPLAY (Guest Storage - z/Architecture)

- SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CLeAr option
  - SET MACHine (to a different machine architecture)
14. The J operand is useful in an adjunct context to retrieve memory from the principal configuration. For example, the following line

```
pgmold=Diag(8,'CPU * DISPLAY RJ140.10')
```

can be included in a REXX exec to capture the z/Architecture program old PSW.

15. For more information, see [“Usage Notes” on page 331](#).

## Responses

### Notes:

1. The value *hexloc* is a hexadecimal storage address.
2. The character displayed for the variable *t* in the string *thexloc* indicates the type of storage:
  - R indicates real storage (second-level storage).
  - V indicates virtual storage (third-level storage).
3. In responses that include third-level storage, an *Rhexloc* trailer shows the second-level address that corresponds to the third-level address. The trailer is displayed on the first line of a range and at every page boundary.

Response 1:

```
thexloc word1 word2 word3 word4 [key] [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the T or N option. The translation is given only for the T option and is in either EBCDIC or (if the TX option is specified) ASCII. The storage key is displayed on the first line of a request, and at every page boundary. If the high-order 32 bits of *hexloc* are not zero, *hexloc* will be displayed as a 64-bit address, in which case the output could wrap to the next line. To make the display more readable, use the M option described in Response 2.

Response 2:

```
thexlochi_ Storage Key - [key]  
hexloclo word1 word2 word3 word4 [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the M option. The translation is displayed in either EBCDIC or (if the MX option is specified) ASCII. The storage key is displayed on the first line, and at every page boundary.

Response 3:

```
thexloc *TRANSLATION* [key] [Rhexloc]
```

is the response you receive for the U option. The value *hexloc* is the location of the first data byte to be displayed. A minimum of 1 character and a maximum of 32 characters are displayed on each line. The contents of storage are displayed only in translated form and are in either EBCDIC or (if the UX option is specified) ASCII. Periods in your display represent unprintable characters. The storage key is displayed on the first line of a request and at every page boundary.

Response 4:

```
thexloc mnemonic x-x mnemonic x-x mnemonic x-x [key] [Rhexloc]
```

is the response you receive for the I option. The value *hexloc* is the location of the first instruction to be displayed. The value *mnemonic* is a character representation of the instruction code; this is a 7-character field in which the mnemonic is left-justified. The expression *x-x* is the actual hexadecimal contents of



storage and is 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. A minimum of 1 instruction and a maximum of 3 instructions are displayed on each line. The storage key is displayed on the first line of a request and at every page boundary.

Response 5:

```
thexloc stringhexadecimaldata [key] [Rhexloc]
```

is the response you receive for the S option. The storage key is displayed on the first line of a request and at every page boundary.

Response 6:

```
thexloc1 TO hexloc2 KEY = key STATE = uc
```

is the response you receive for the K option. The values *hexloc1* and *hexloc2* are storage locations, *key* is the associated storage key for *hexloc1*, and *uc* is a two-letter code describing the block-usage and block-content state:

**u**

is a code for the block-usage state:

**Code**

**Meaning**

**S**

Stable

**U**

Unused

**V**

Volatile

**P**

Potentially volatile

**c**

is a code for the block-content state:

**Code**

**Meaning**

**R**

Resident

**P**

Preserved

**Z**

Logically zero

For more information about block-usage and block-content states, see [Collaborative memory management assist in z/VM: CP Programming Services](#).

A response is sent for every contiguous address range that has unique storage keys and whenever a segment boundary is crossed.

Response 7:

The PSW is used to resolve a logical address. A PSW that is not valid causes the current request to be rejected and processing to continue with the next request (if any) on the command line. In all cases, a reset PSW (all-zero) is accepted. The rejected operands are displayed along with a response indicating that the PSW is not valid. The response is:

```
addr is not addressable because of an invalid PSW.
```

## DISPLAY (Guest Storage - z/Architecture)

where *addr* is the rejected operand or operands; it specifies an address space and a storage location within that address space.

Response 8:

The dynamic-address translation (DAT) process uses the segment table designator (ASCE) to convert a third-level address to a second-level address (storage that appears real to your virtual machine). The translation process may generate one of several exceptions, identifying why one or more pages of storage are not addressable. A range of rejected storage addresses is displayed along with a response indicating the nature of the exception condition. The responses are:

```
thexloc1 to hexloc2 non-addressable storage  
thexloc1 to hexloc2 non-addressable storage  
- protection exception  
thexloc1 to hexloc2 non-addressable storage  
- addressing exception  
thexloc1 to hexloc2 non-addressable storage  
- segment translation exception  
thexloc1 to hexloc2 non-addressable storage  
- page translation exception  
thexloc1 to hexloc2 non-addressable storage  
- translation specification exception  
thexloc1 to hexloc2 non-addressable storage  
- ASCE type exception  
thexloc1 to hexloc2 non-addressable storage  
- region-first-translation exception  
thexloc1 to hexloc2 non-addressable storage  
- region-second-translation exception  
thexloc1 to hexloc2 non-addressable storage  
- region-third-translation exception  
thexloc1 to hexloc2 non-addressable storage  
- block volatility exception
```

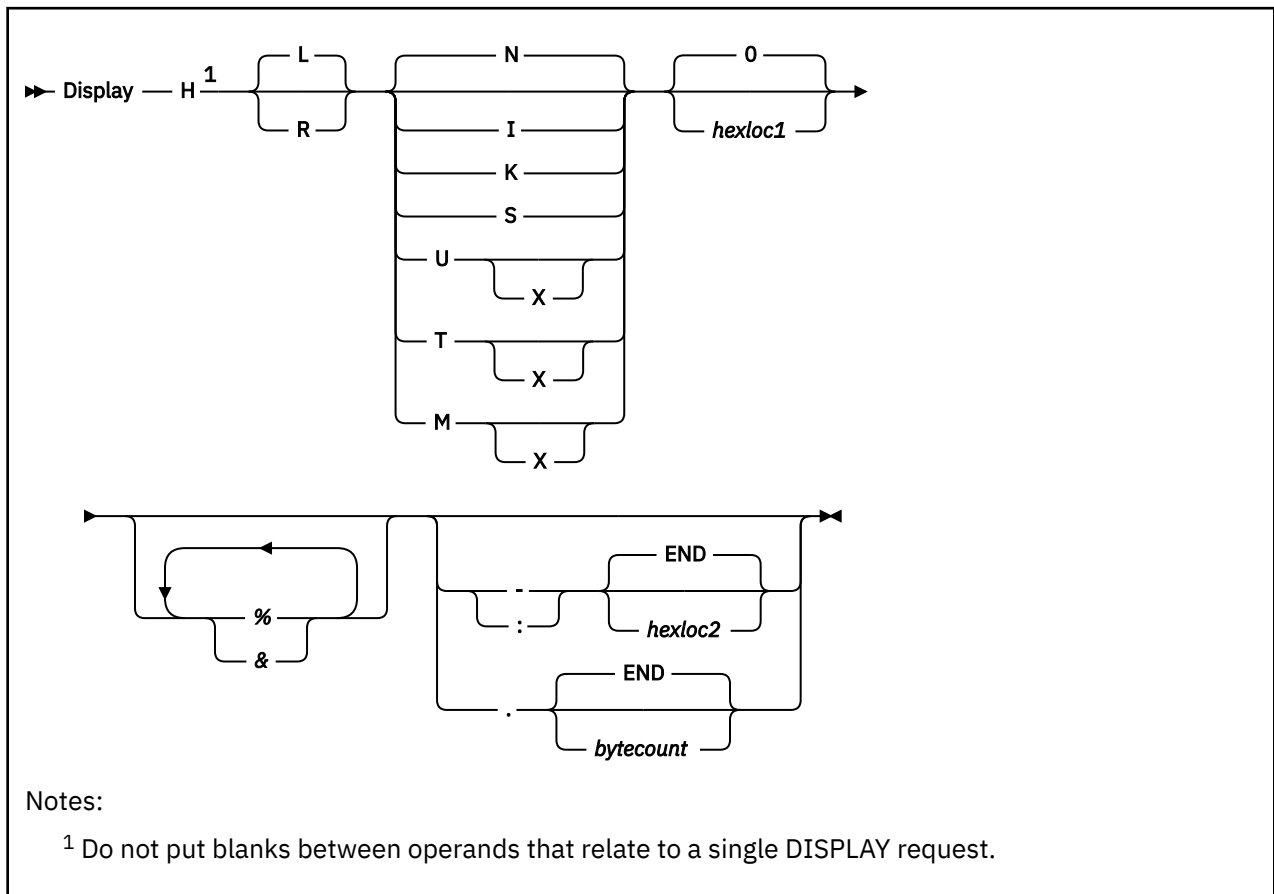
where *hexloc1* is the starting address and *hexloc2* the ending address of the non-addressable storage. The addresses take into account basing, indexing, and wrapping and are padded to 8 digits.

Response 9:

If your display request includes a request to display identical lines, you receive the following response:

```
thexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...
```

## DISPLAY (Host Storage)



### Authorization

Privilege Class: C, E

### Purpose

Use DISPLAY (Host Storage) to see the contents of host (first-level) storage.

### Operands

**L**

specifies that host logical storage addresses are to be displayed. Host logical storage consists of 4 KB pages where each page boundary has an address that ends in X'000'. This is the default.

**R**

specifies that host real storage addresses are to be displayed. Host real storage consists of 4 KB frames where each frame boundary has an address that ends in X'000'.

**N**

displays the contents of storage in hexadecimal without character translation. This is the default. Storage is displayed in word segments. An address range is displayed starting at the fullword in which the specified address resides. Storage is displayed in fullword increments.

The storage key is included on the first line and at every page or frame boundary.

## DISPLAY (Host Storage)

### I

displays the contents of storage in instruction format. The specified address is rounded down to the nearest halfword boundary.

The storage key is included on the first line and at every page or frame boundary.

### K

displays in hexadecimal the storage keys for all frames or partial frames included in the specified address range. If you are displaying host logical storage addresses, these are storage keys for the backing frames in host real storage.

### S

displays the contents of storage in hexadecimal translation. Storage is displayed as a string starting at the location specified. The specified address range is not boundary-aligned.

The storage key is included on the first line and at every page or frame boundary.

### T

displays the contents of storage in hexadecimal and character form. Storage is displayed in word segments. An address range is displayed starting at the X'10'-byte boundary in which the specified address resides. X'10' bytes of storage are displayed per line. The default display is in EBCDIC; the display is in ASCII if the X option is specified.

The storage key is included on the first line and at every page or frame boundary.

### U

displays the contents of storage in character form. Storage is displayed starting at the address specified as an EBCDIC string or as an ASCII string if the X option is specified. The specified address range is not boundary-aligned.

The storage key is included on the first line and at every page or frame boundary.

### M

displays the contents of storage in hexadecimal and character form. A header is included as the first line and at every page or frame boundary. It contains the high-order 32 bits of the 64-bit address along with the storage key. Storage is displayed in word segments. An address range is displayed starting at the X'10' byte boundary in which the specified address resides. X'10' bytes of storage are displayed per line. The hexadecimal display is followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

### O

#### ***hexloc1***

is the starting storage location that is to be displayed. The address *hexloc1* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The default is zero.

Storage locations and the minimum amount displayed are rounded according to the following conventions (in hexadecimal):

Option	Start	Minimum Length
T	10	10
M	10	10
K	800	800
S	1	1
I	2	2 - 6 depending on the instruction
N	4	4
U	1	1

%

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the DISPLAY command.

&amp;

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the DISPLAY command.

**-hexloc2****:hexloc2****-END****:END**

is the last of the range of hexadecimal storage locations whose contents are to be displayed. If specified, *hexloc2* must be equal to or greater than *hexloc1* and must be within the bounds of addressable storage (see usage note “3” on page 365). The address *hexloc2* may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. A colon (:) means the same as a dash (-) when you are specifying a range of addresses.

Specify END if you want to display the remaining contents of storage. END is the default if you specify a dash (-) or a colon (:) without an address.

**.bytecount****.END**

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at *hexloc1*) to be displayed. If specified, the *bytecount* must have a value of at least 1 and may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The sum of *hexloc1* and *bytecount* must be an address within the bounds of addressable storage (see usage note “3” on page 365).

Specify END if you want to display the remaining contents of storage. END is the default if you specify a period (.) without a byte count value.

## Usage Notes

1. While host storage is being displayed, the running of a virtual machine or someone else using the STORE (Host Storage) command could cause data to be changed. This could create an inconsistency in the data that is viewed.
2. In a multiprocessor environment, each processor has a prefix register that it uses to relocate addresses between 0 and X'1FFF' (an 8 KB area known as the prefix area) to two other contiguous frames in real storage. The prefix register enables each processor to use a different prefix area to avoid conflict with other processors for such activity as interrupt code recording. Thus, the range 0 through X'1FFF' refers to different areas of storage, depending on which processor generates the address. All references to first-level storage are handled as if they were made on the IPL processor. Because those locations in the system execution space are identity-mapped, addresses in the range 0 through X'1FFF' have the same content in DISPLAY HL output and DISPLAY HR output.
3. For host logical storage (the system execution space), addressable storage includes the system execution area, which resides below 2 GB, plus the system frame table and the system execution space page management table, which reside above 2 GB. To display the storage address ranges used for the system execution space, issue the QUERY SXSSTORAGE command. The address range that you specify on the DISPLAY command must start and end in the same address range of the system execution space. If the END operand is specified (or allowed to default), it refers to the end of the range in which the starting address is located.

For host real storage, addressable storage is the lesser of CP's configured storage size (if specified at system generation) and the real machine storage size. To display the size of host real storage, issue the QUERY STORAGE command.

4. When displaying host logical storage, if you specify an address that is not backed with a real storage frame, error message HCP6151E is issued, even if the address is included in the ranges of in-use and usable addresses displayed by the QUERY SXSSTORAGE command.
5. If you use an underscore, it must be followed by 8 hexadecimal digits.
6. For more information, see [“Usage Notes” on page 331](#).

## Responses

### Notes:

1. The variable *hexloc* represents a hexadecimal storage address. The variables *hexlochi* and *hexloclo* represent the high-order 32 bits and low-order 32 bits of a 64-bit hexadecimal storage address.
2. The character displayed for the variable *t* in the string *Hthexloc* indicates the type of storage:
  - L indicates host logical storage.
  - R indicates host real storage.
3. In responses that include host logical storage, an *Rhexloc* trailer shows the host real storage address that corresponds to the host logical storage address. The trailer is displayed on the first line of a range and at every page boundary.

### Response 1:

```
Hthexloc word1 word2 word3 word4 [key] [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the T or N option. The translation is displayed only if you have also entered the T option and is in either EBCDIC or (if the TX option is specified) ASCII. The storage key is displayed on the first line, and at every page or frame boundary.

### Response 2:

```
Hthexlochi_ Storage Key - [key]  

hexloclo word1 word2 word3 word4 [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the M option. The translation is displayed in either EBCDIC or (if the MX option is specified) ASCII. The storage key is displayed on the first line, and at every page or frame boundary.

### Response 3:

```
Hthexloc *TRANSLATION* [key] [Rhexloc]
```

is the response you receive for the U option. The value *hexloc* is the location of the first data byte to be displayed. A minimum of 1 character and a maximum of 32 characters is displayed on each line. The contents of storage are displayed in translated form and are in either EBCDIC or (if the UX option is specified) ASCII. Periods in the display data represent unprintable characters. The storage key is displayed on the first line, and at every page or frame boundary.

### Response 4:

```
Hthexloc mnemonic x-x mnemonic x-x mnemonic x-x [key] [Rhexloc]
```

is the response you receive for the I option. The value *hexloc* is the location of the first instruction to be displayed. The value *mnemonic* is a character representation of the instruction code; this is a 7-character field in which the mnemonic is left-justified. The expression *x-x* is the actual hexadecimal contents of storage and is 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. A minimum of 1 instruction and a maximum of 3 instructions are displayed on each line. The storage key is displayed on the first line and at every page or frame boundary.

### Response 5:

```
Hthexloc stringhexadecimaldata [key] [Rhexloc]
```

is the response you receive for the S option. The storage key is displayed on the first line of a range, and at every page or frame boundary.

Response 6:

```
Hthexloc1 TO hexloc2 KEY = key
```

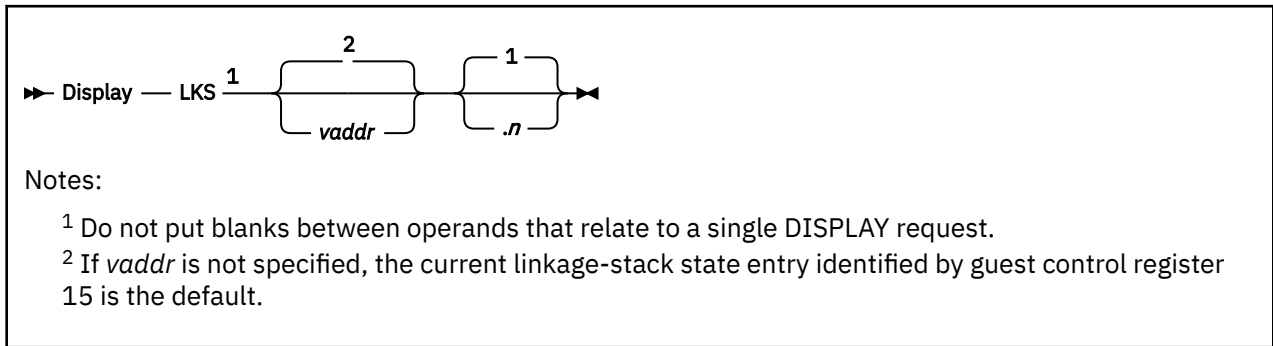
is the response you receive for the K option. The values *hexloc1* and *hexloc2* are storage locations, and *key* is the associated storage key for *hexloc1*. You receive this response for every contiguous address range that has unique storage keys, and whenever a segment boundary is crossed.

Response 7:

If your display request includes a request to display one or more lines identical with the previous line, you receive the following response:

```
Hthexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...
```

## DISPLAY LKS



### Authorization

Privilege Class: G

### Purpose

Use DISPLAY LKS to see the state entries of a guest linkage stack.

### Operands

#### *vaddr*

displays the branch or program-call linkage-stack state entry that contains the specified guest virtual address (*vaddr*) within the current home address space designated by guest control register 13. The default is the current linkage-stack state entry identified by guest control register 15.

**1**

***n***

*n* is a decimal number, up to eight digits, specifying the number of branch and program-call linkage-stack state entries to be displayed. The display starts with the specified/current entry and moves back toward the first entry on the linkage stack. 1 is the default, and 0 is treated as 1.

### Usage Notes

1. This command is valid only for an ESA virtual machine.
2. The *first* linkage-stack state entry is the oldest state entry (the first one put on the stack). The *last* linkage-stack state entry is the current state entry (the most recent one put on the stack).
3. Unlike other DISPLAY command operands, Linkage Stack does not define the default for subsequent operands of the current DISPLAY function.
4. The virtual address specified in *vaddr* must fall within the branch or program-call linkage-stack state entry, including the first and last bytes of the linkage-stack state entry.
5. A linkage stack state entry is an entry on the linkage stack that describes the environment in an address space (such as access registers, general-purpose registers, and PSWs).
6. For more information, see [“Usage Notes” on page 331](#).

### Responses

Heading	Description
ED-ADDR	The linkage-stack-entry description address, a guest home virtual address consisting of eight hexadecimal digits (16 hexadecimal digits for a z/Architecture guest).



Heading	Description
U	The unstack-suppression bit, either 0 or 1.
ET	The entry type, either BRN for a branch state entry or PRC for a program-call state entry.
SI	The section identification, represented as two hexadecimal digits.
RFS	The remaining free space, represented as four hexadecimal digits.
NES	The next-entry size, represented as four hexadecimal digits.
PKM	The PSW-key mask, represented as four hexadecimal digits.
SASN	The secondary ASN, represented as four hexadecimal digits.
EAX	The extended authorization index, represented as four hexadecimal digits.
PASN	The primary ASN, represented as four hexadecimal digits.
PSW	The program-status word, represented as 16 hexadecimal digits (32 hexadecimal digits for a z/Architecture guest).
A	The addressing mode, which may be either a 0 or 1 for an ESA/390 guest and either 24, 31 or 64 for a z/Architecture guest.
B-ADDR	The branch address, represented as eight hexadecimal digits (16 hexadecimal digits for a z/Architecture guest).
MOD-AREA	The modifiable area, represented as 16 hexadecimal digits.
PC-NUM	The program call number, represented as eight hexadecimal digits.
CSID	The called-space identification, if any, represented as eight hexadecimal digits.
SINST	Secondary ASTE instance number, if any, represented as eight hexadecimal digits.
PINST	Primary ASTE instance number, if any, represented as eight hexadecimal digits.
GPR	General purpose register, represented as eight hexadecimal digits.
GRG	General purpose register for z/Architecture guests, represented as 16 hexadecimal digits.
AR	Access register, represented as eight hexadecimal digits.

Response 1:

The linkage-stack entry for an ESA/390 branch state entry has the following format:

ED-ADDR	U	ET	SI	RFS	NES	PKM	SASN	EAX	PASN
nnnnnnnn	n	BRN	nn	nnnn	nnnn	nnnn	nnnn	nnnn	nnnn
PSW		A	B-ADDR	MOD-AREA					
nnnnnnnn	nnnnnnnn	n	nnnnnnnn	nnnnnnnnnnnnnnnnnn					
GPR	0 =	greg0	greg1	greg2	greg3				
GPR	4 =	greg4	greg5	greg6	greg7				
GPR	8 =	greg8	greg9	greg10	greg11				
GPR	12 =	greg12	greg13	greg14	greg15				
AR	0 =	areg0	areg1	areg2	areg3				
AR	4 =	areg4	areg5	areg6	areg7				
AR	8 =	areg8	areg9	areg10	areg11				
AR	12 =	areg12	areg13	areg14	areg15				

The headings in this response are described under [“Responses”](#) on page 368.

Response 2:

The linkage-stack entry for a z/Architecture branch state entry has the following format:

```

ED-ADDR      U ET  SI RFS  NES  PKM  SASN  EAX  PASN
nnnnnnnnnnn n BRN nn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn

                PSW                      A      B-ADDR
nnnnnnnnnnn  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nn  nnnnnnnnnnnnnnnnnnn

        MOD-AREA      SINST      PINST
nnnnnnnnnnnnnnnnnnn nnnnnnnnnn nnnnnnnnnn

GRG  0 = greg0                greg1
GRG  2 = greg2                greg3
GRG  4 = greg4                greg5
GRG  6 = greg6                greg7
GRG  8 = greg8                greg9
GRG 10 = greg10               greg11
GRG 12 = greg12               greg13
GRG 14 = greg14               greg15

AR   0 = areg0  areg1  areg2  areg3
AR   4 = areg4  areg5  areg6  areg7
AR   8 = areg8  areg9  areg10 areg11
AR  12 = areg12 areg13 areg14 areg15
    
```

The headings in this response are described under [“Responses” on page 368](#).

ED-ADDR and B-ADDR can be either 32 or 64 bits.

Response 3:

The linkage-stack entry for an ESA/390 program call state entry has the following format:

```

ED-ADDR  U ET  SI RFS  NES  PKM  SASN  EAX  PASN
nnnnnnnn n PRC nn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn

                PSW      PC-NUM      MOD-AREA      CSID
nnnnnnnnnnn  nnnnnnnnnn nnnnnnnnnnnnnnnnnnn nnnnnnnnnn

GPR  0 = greg0  greg1  greg2  greg3
GPR  4 = greg4  greg5  greg6  greg7
GPR  8 = greg8  greg9  greg10 greg11
GPR 12 = greg12 greg13 greg14 greg15

AR   0 = areg0  areg1  areg2  areg3
AR   4 = areg4  areg5  areg6  areg7
AR   8 = areg8  areg9  areg10 areg11
AR  12 = areg12 areg13 areg14 areg15
    
```

The headings in this response are described under [“Responses” on page 368](#).

Response 4:

The linkage-stack entry for a z/Architecture program call state entry has the following format:

```

ED-ADDR      U ET  SI RFS  NES  PKM  SASN  EAX  PASN
nnnnnnnnnnn n PRC nn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn

                PSW                      PC-NUM
nnnnnnnnnnn  nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn

        MOD-AREA      CSID      SINST      PINST
nnnnnnnnnnnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn

GRG  0 = greg0                greg1
GRG  2 = greg2                greg3
GRG  4 = greg4                greg5
GRG  6 = greg6                greg7
GRG  8 = greg8                greg9
GRG 10 = greg10               greg11
GRG 12 = greg12               greg13
GRG 14 = greg14               greg15

AR   0 = areg0  areg1  areg2  areg3
AR   4 = areg4  areg5  areg6  areg7
    
```

```
AR 8 = areg8 areg9 areg10 areg11
AR 12 = areg12 areg13 areg14 areg15
```

The headings in this response are described under [“Responses”](#) on page 368.

ED-ADDR can be either 32 or 64 bits.

Response 5:

If a guest virtual address from the home address space cannot be converted to a host address, the translation process generates one of several exceptions, identifying the first page that was not addressable. The responses are:

```
Vhexloc1 to hexloc2 nonaddressable storage
- protection exception
Vhexloc1 to hexloc2 nonaddressable storage
- addressing exception
Vhexloc1 to hexloc2 nonaddressable storage
- segment translation exception
Vhexloc1 to hexloc2 nonaddressable storage
- page translation exception
Vhexloc1 to hexloc2 nonaddressable storage
- translation specification exception
```

The operand *Vhexloc1* is the starting address, and *hexloc2* the ending address, of the nonaddressable storage.

Response 6:

If a state entry that is neither a branch nor a program call is reached, the following response is issued:

```
Vhexloc1 to hexloc2 are linkage stack
state entries that are not branch or program-call.
```

The operand *Vhexloc1* is the starting address, and *hexloc2* the ending address, of the block of contiguous state entries in a linkage stack section that are not a branch or a program call.

Response 7:

```
The linkage stack is empty.
```

is issued if the linkage stack is empty.

Response 8:

If the specified guest virtual address is not within a branch or program-call state entry, the following response is issued:

```
Virtual address hexloc is not within a linkage
stack branch state entry or a program-call state entry.
```

The operand *hexloc* is the guest virtual address specified on the command line. Header and trailer state entries are not displayed.

Response 9:

```
The first state entry in the linkage stack has been reached.
```

is issued if the bottom of the stack is reached during command processing.

## DISPLAY PREFIX

---

```
►► Display — PREFIX ◄◄
```

### Authorization

Privilege Class: G

### Purpose

Use DISPLAY PREFIX to display the contents of the prefix register. This shows you the absolute location of guest page 0.

### Usage Notes

1. For usage information, see [“Usage Notes” on page 331](#).

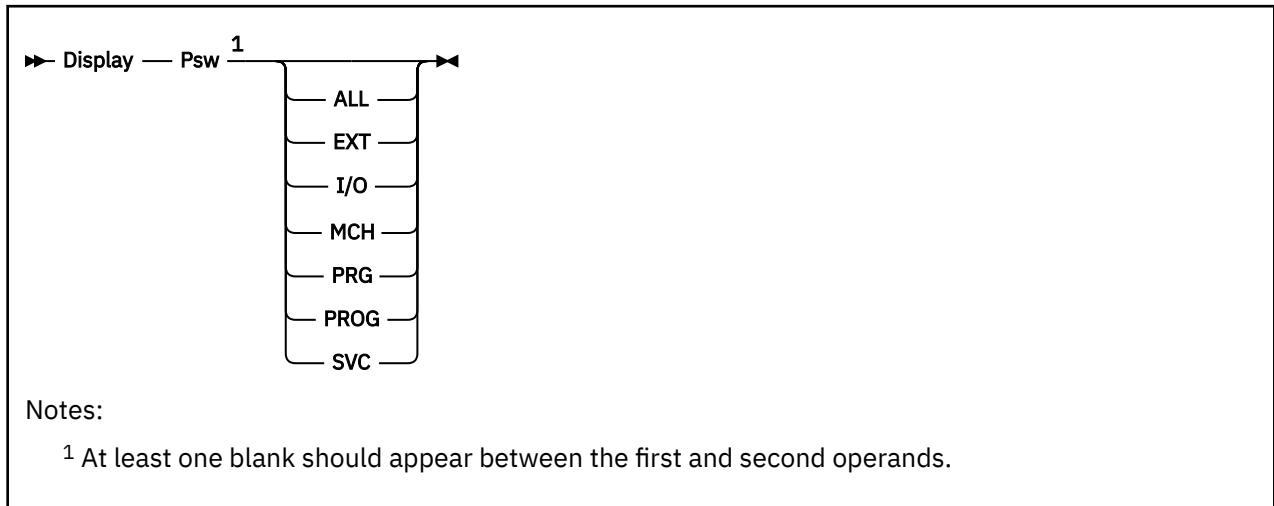
### Responses

Response 1:

```
PREFIX = xxxxxxxx
```

displays in hexadecimal the contents of the prefix register.

## DISPLAY PSW



### Authorization

Privilege Class: G

### Purpose

Use DISPLAY PSW to see the program status word of your virtual machine.

### Operands

#### ALL

displays the current values and interruption code information for the current PSW and all old and new interrupt PSWs (except RESTART).

#### EXT

#### I/O

#### MCH

#### PRG

#### PROG

#### SVC

displays the current values and appropriate interruption code information for both the new and old PSWs specified—that is, external, SVC, program check, machine check, and I/O.

### Usage Notes

1. The DISPLAY PSW command is valid for a z/Architecture or z/XC virtual machine only if SET PSWTRANS (with ALL or DISPLAY option) has been issued previously.
2. DISPLAY PSW ALL displays the PSW as a 128-bit PSW for z/Architecture and z/XC virtual machines.
3. For a z/Architecture or z/XC virtual machine, the PSW is 128 bits to accommodate 64-bit instruction addresses. An expanded PSW is labeled in the command output as "PSWG". The SET PSWTRANS command allows the PSWs from a z/Architecture or z/XC virtual machine to be shown and manipulated in the 64-bit ESA/390 format.

When the DISPLAY PSW command is issued from a z/Architecture or z/XC virtual machine (with a previous SET PSWTRANS ALL/DISPLAY issued), the 128-bit z/Architecture format PSW must be converted to a 64-bit ESA/390 format PSW. This conversion is done as follows:

## DISPLAY PSW

- bits 0-31 are set to bits 0-31 of the z/Architecture or z/XC PSW with bit 12 set to 1 and bit 31 set to 0.
- bits 33-63 of the PSW are set to bits 97-127 of the z/Architecture or z/XC PSW. If bits 64-95 of the z/Architecture or z/XC PSW are not 0, the PSW is flagged as non-translatable.
- bit 32 is set as follows:
  - if bits 31 and 32 of the z/Architecture or z/XC PSW are both 1, the PSW is flagged as non-translatable
  - if bits 31 and 32 of the z/Architecture or z/XC PSW are both 0, bit 32 of the ESA/390 PSW is set to 0
  - if bit 31 of the z/Architecture or z/XC PSW is 0 and bit 32 is 1, bit 32 of the ESA/390 PSW is set to 1.

## Responses

### Response 1:

The command DISPLAY PSW with no other options displays the current PSW. This is the response you receive on ESA/390 and ESA/XC virtual machines:

```
PSW = 03E40000 8003010C
```

### Response 2:

The command DISPLAY PSW with no other options displays the current PSW. This is the response you receive on z/Architecture and z/XC virtual machines:

```
PSW = 03EC0000 8003010C PSWG = 03E40000 80000000 00000000 0003010C
```

### Response 3:

The command DISPLAY PSW with no other options displays the current PSW. If the current PSW for a guest machine cannot be converted to a 64-bit PSW, then this is the response that you receive on z/Architecture and z/XC virtual machines:

```
PSW = NON TRANSLATABLE PSWG = 03E40000 80000000 00000000 0003010C
```

### Response 4:

The command DISPLAY PSW ALL displays the old and new PSWs and expands the interruption code into four hexadecimal digits in front of the old PSWs. *nnnn* represents the EC-mode interruption code. The numbers 18, 20, 28... are storage locations. Each *xxxxxxxx xxxxxxxx* represents the content of a PSW. This is the response you receive on ESA/390 and ESA/XC virtual machines:

```
PSW = xxxxxxxx xxxxxxxx  
EXT nnnn 18 OLD xxxxxxxx xxxxxxxx 58 NEW xxxxxxxx xxxxxxxx  
SVC nnnn 20 OLD xxxxxxxx xxxxxxxx 60 NEW xxxxxxxx xxxxxxxx  
PRG nnnn 28 OLD xxxxxxxx xxxxxxxx 68 NEW xxxxxxxx xxxxxxxx  
MCH nnnn 30 OLD xxxxxxxx xxxxxxxx 70 NEW xxxxxxxx xxxxxxxx  
I/O nnnn 38 OLD xxxxxxxx xxxxxxxx 78 NEW xxxxxxxx xxxxxxxx
```

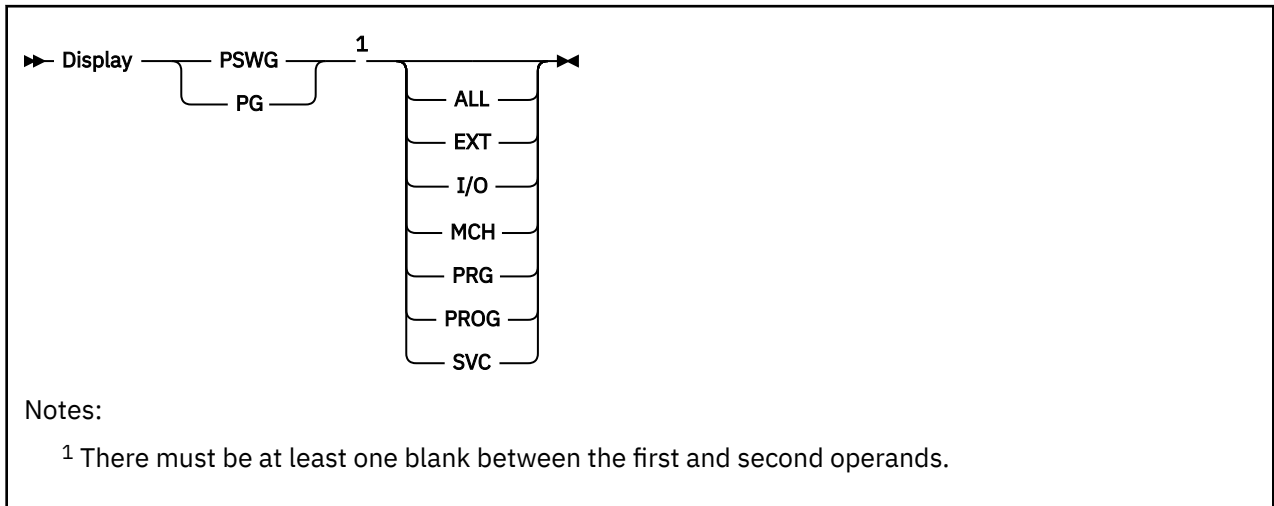
### Response 5:

The command DISPLAY PSW ALL displays the old and new PSWs and expands the interruption code into four hexadecimal digits in front of the old PSWs. *nnnn* represents the interruption code. The numbers 130, 140, 150, ... are storage locations. Each *xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx* represents the content of a PSW. This is the response that you receive on z/Architecture and z/XC virtual machines.

```
PSW = 03EC0000 8003010C PSWG = 03E40000 80000000 00000000 0003010C  
EXT nnnn 130 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  
1B0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  
SVC nnnn 140 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  
1C0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  
PRG nnnn 150 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

```
1D0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  
MCH nnnn 160 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  
1E0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  
I/O nnnn 170 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  
1F0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

## DISPLAY PSWG



### Authorization

Privilege Class: G

### Purpose

Use DISPLAY PSWG to see the program status word of your z/Architecture or z/XC virtual machine.

### Operands

#### ALL

displays the current values and interruption code information for the current PSW and all old and new interrupt PSWs (except RESTART).

#### EXT

#### I/O

#### MCH

#### PRG

#### PROG

#### SVC

displays the current values and appropriate interruption code information for both the new and old PSWs specified—that is, external, SVC, program check, machine check, and I/O.

### Usage Notes

- To display the PSW for an ESA/390 or ESA/XC virtual machine, use the DISPLAY PSW command.

### Responses

Response 1:

The command DISPLAY PSWG with no other options displays the current PSW.

```
PSW = 03E40000 80000000 00000000 0003010C
```

Response 2:

The command DISPLAY PSWG ALL displays the old and new PSWs and expands the interruption code into four hexadecimal digits in front of the old PSWs. *nnnn* represents the interruption code. The numbers 130,



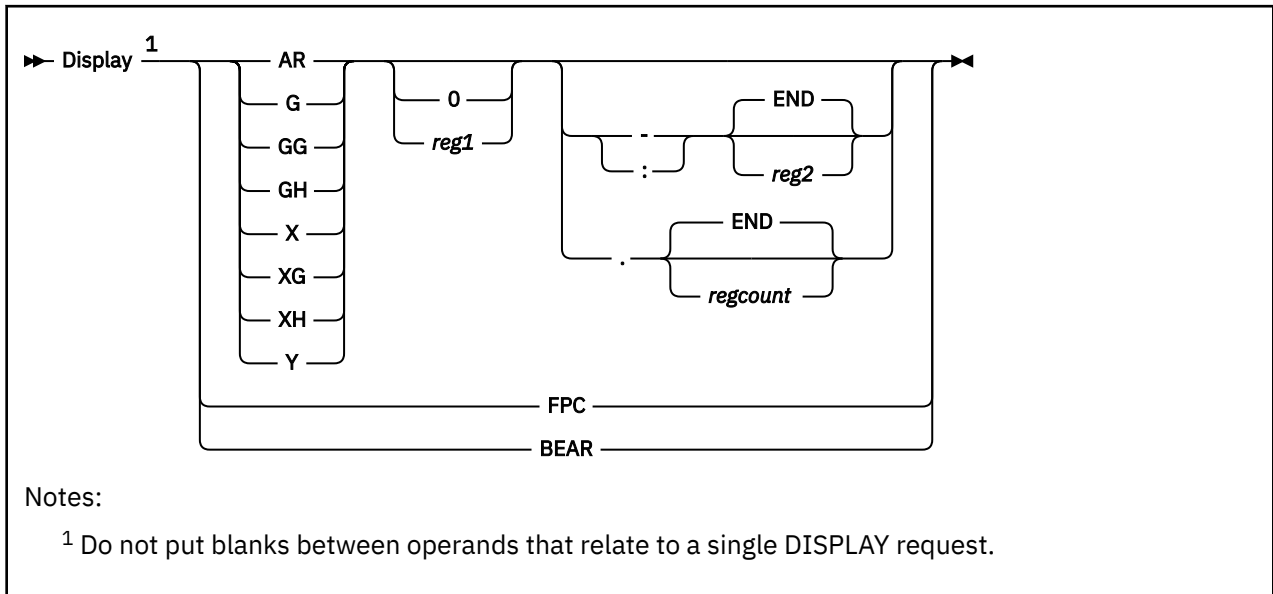
140, 150, ... are storage locations. Each xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx represents the content of a PSW.

```

PSW = 03E40000 80000000 00000000 0003010C
EXT nnnn 130 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          1B0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
SVC nnnn 140 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          1C0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
PRG nnnn 150 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          1D0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
MCH nnnn 160 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          1E0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
I/O nnnn 170 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          1F0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

```

## DISPLAY (Registers)



### Authorization

Privilege Class: G

### Purpose

Use DISPLAY (Registers) to see the following:

- Access registers
- General-purpose registers
- Control registers.
- Floating-point registers
- Floating-point control register
- Breaking-event-address register

### Operands

#### ARreg1

*reg1* is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, access register whose contents are to be displayed. If you specify AR without a register number, the contents of all access registers are displayed.

#### Greg1

G indicates a full register for an ESA/390 or ESA/XC guest or the low half of the register for a z/Architecture or z/XC guest. *reg1* is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be displayed. If you specify G without a register number, the appropriate contents of all of the general registers are displayed.

#### GGreg1

GG indicates a full register (32 bits for an ESA/390 or ESA/XC guest, 64 bits for a z/Architecture or z/XC guest). *reg1* is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be displayed. If you specify GG without a register number, the appropriate contents of all of the general registers are displayed.

**GHreg1**

GH indicates the upper half of a register for a z/Architecture or z/XC guest. This option is not valid for an ESA/390 or ESA/XC guest. *reg1* is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be displayed. If you specify GH without a register number, the high halves of all of the general registers are displayed.

**Xreg1**

X indicates a full control register for an ESA/390 or ESA/XC guest or the low half of the control register for a z/Architecture or z/XC guest. *reg1* is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, control register whose contents are to be displayed. If you specify X without a register number, the appropriate contents of all of the control registers are displayed.

**XGreg1**

XG indicates a full control register (32 bits for an ESA/390 or ESA/XC guest, 64 bits for a z/Architecture or z/XC guest). *reg1* is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, control register whose contents are to be displayed. If you specify XG without a register number, the appropriate contents of all of the control registers are displayed.

**XHreg1**

XH indicates the upper half of a control register for a z/Architecture or z/XC guest. This option is not valid for an ESA/390 or ESA/XC guest. *reg1* is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, control register whose contents are to be displayed. If you specify XH without a register number, the high halves of all of the control registers are displayed.

**Yreg1**

*reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, floating-point register whose contents are to be displayed. If you specify Y without a register number, the contents of all the floating-point registers are displayed.

**FPC**

FPC displays the contents of the floating-point control register.

**BEAR**

BEAR displays the contents of the breaking-event-address register if the PER-3 Facility is available.

**: END****- reg2**

*reg2* is a number of the last register whose contents are to be displayed. You must specify the dash (-) or the colon (:) when you are requesting more than one register by register number. You must specify the range of registers in ascending order. Therefore, the value you specify for *reg2* must be equal to or greater than the value you specify for *reg1*.

The colon (:) means the same as a dash (-) when you are specifying a range of addresses.

If you specify AR, G, GG, GH, X, XG, XH, or Y, *reg2* may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F. The contents of *reg1* through *reg2* are displayed.

Specify END to display the contents of all remaining registers. END is the default.

**END****. regcount**

*regcount* is a decimal number from 1 to 16 or a hexadecimal number from 1 to F specifying the number of registers (starting with register 1) whose contents are to be displayed. If you specify AR, G, GG, GH, X, XG, XH, or Y, *regcount* can be a decimal number from 1 to 16 or a hexadecimal number from 1 to F.

The sum of *reg1* and *regcount* must be a number that does not exceed the maximum register number for the type of registers being displayed.

Specify END to display the contents of all remaining registers. END is the default.

## Usage Notes

1. The G and X operands are valid for an ESA/390, ESA/XC, z/Architecture, or z/XC guest. For an ESA/390 or ESA/XC guest, these operands refer to the full four-byte register. For a z/Architecture or /XC guest, these operands refer to the low-order four bytes of the full eight-byte register.
2. The GG and XG operands are valid for an ESA/390, ESA/XC, z/Architecture, or z/XC guest. For an ESA/390 or ESA/XC guest, these operands refer to the full four-byte register. For a z/Architecture or z/XC guest, these operands refer to the full eight-byte register.
3. The GH and XH operands are valid only for a z/Architecture or z/XC guest. These operands refer to the high-order four bytes of the full eight-byte register.
4. When a dash (-), colon (:), or period (.) is not preceded by a register number, the first register number defaults to 0. When a dash, colon, or period is not followed by a register number, the last register defaults to END.
5. For more information, see [“Usage Notes” on page 331](#).

## Responses

Response 1:

*Displaying General Registers:*

```
GPR nn = genreg0 genreg1 genreg2 genreg3
```

is the response you receive when you display general registers using *D G* or *D GG* from an ESA/390 or ESA/XC guest. *nn* indicates the first (or only) register whose contents are displayed.

The content of each register is displayed in hexadecimal. Up to four registers per line are displayed for each range you specify. If required, multiple lines are displayed to a maximum of four lines.

Response 2:

*Displaying General Registers:*

```
HCPCDE6150E An option has been specified that is not valid in
ESA/390 mode.
```

is the response you receive when you display general registers using *D GH* from an ESA/390 guest (an appropriate message will be issued from an ESA/XC guest).

Response 3:

*Displaying General Registers:*

```
GPR nn = genreg0l genreg1l genreg2l genreg3l
```

is the response you receive when you display general registers using *D G* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are displayed.

The content of each register is displayed in hexadecimal and represents the low half of the full 64-bit register. Up to four registers per line are displayed for each range you specify. If required, multiple lines are displayed to a maximum of four lines.

Response 4:

*Displaying General Registers:*

```
GRG nn = genreg0hgenreg0l genreg1hgenreg1l
```

is the response you receive when you display general registers using *D GG* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are displayed.

The content of each register is displayed in hexadecimal and represents the the full 64-bit register. Up to two registers per line are displayed for each range you specify. If required, multiple lines are displayed to a maximum of eight lines.

The high and low halves of each register may be separated by an optional underscore for readability. To have the underscore displayed, use the SET UNDERSCORE command.

Response 5:

*Displaying General Registers:*

```
GRH nn = genreg0h genreg1h genreg2h genreg3h
```

is the response you receive when you display general registers using *D GH* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are displayed.

The content of each register is displayed in hexadecimal and represents the high half of the full 64-bit register. Up to four registers per line are displayed for each range you specify. If required, multiple lines are displayed to a maximum of four lines.

Response 6:

*Displaying Control Registers:*

```
ECR nn = ctlreg0 ctlreg1 ctlreg2 ctlreg3
```

is the response you receive when you display control registers using *D X* or *D XG* from an ESA/390 or ESA/XC guest. *nn* indicates the first (or only) register whose contents are displayed.

The content of each register is displayed in hexadecimal. Up to four registers per line are displayed for each range you specify. If required, multiple lines are displayed to a maximum of four lines.

Response 7:

*Displaying Control Registers:*

```
HCPCDE6150E An option has been specified that is not valid in
ESA/390 mode.
```

is the response you receive when you display control registers using *D XH* from an ESA/390 guest (an appropriate message will be issued from an ESA/XC guest).

Response 8:

*Displaying Control Registers:*

```
ECR nn = ctlreg0l ctlreg1l ctlreg2l ctlreg3l
```

is the response you receive when you display control registers using *D X* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are displayed.

The content of each register is displayed in hexadecimal and represents the low half of the full 64-bit register. Up to four registers per line are displayed for each range you specify. If required, multiple lines are displayed to a maximum of four lines.

Response 9:

*Displaying Control Registers:*

```
CRG nn = ctlreg0hctlreg0l ctlreg1hctlreg1l
```

is the response you receive when you display control registers using *D XG* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are displayed.

The content of each register is displayed in hexadecimal and represents the the full 64-bit register. Up to two registers per line are displayed for each range you specify. If required, multiple lines are displayed to a maximum of eight lines.

## DISPLAY (Registers)

The high and low halves of each register may be separated by an optional underscore for readability. To have the underscore displayed, use the SET UNDERSCORE command.

Response 10:

*Displaying Control Registers:*

```
CRH nn = ctlreg0h ctlreg1h ctlreg2h ctlreg3h
```

is the response you receive when you display control registers using *D XH* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are displayed.

The content of each register is displayed in hexadecimal and represents the high half of the full 64-bit register. Up to four registers per line are displayed for each range you specify. If required, multiple lines are displayed to a maximum of four lines.

Response 11:

*Displaying Access Registers:*

```
AR nn = areg0 areg1 areg2 areg3
```

is the response you receive when you display access registers; *nn* is the first (or only) access register whose contents are displayed.

The contents of the requested access registers are displayed in hexadecimal. Up to four registers per line are displayed. If required, multiple response lines are displayed to a maximum of four lines.

Response 12:

*Displaying Floating Point Registers:*

```
FPR nn = xxxxxxxxxxxxxxxx ±.ddddddddddddddd E dd
```

is the response you receive when you display floating-point registers. The *nn* is the floating-point register whose contents are displayed on the line. The contents of the floating-point registers are displayed in both the internal hexadecimal format and the E-format; the first part of the response is the contents of the register in hexadecimal, and the second part of the response is the contents in exponential decimal format. Only one register is displayed per line. If you specified a range of registers, multiple response lines are displayed.

Response 13:

*Displaying Floating-Point Control Register:*

```
FPC = mmmmmmm
```

is the response you receive when you display the floating-point control register. The contents of the floating-point control register are displayed in hexadecimal format.

Response 14:

*Displaying Breaking-event-address Register:*

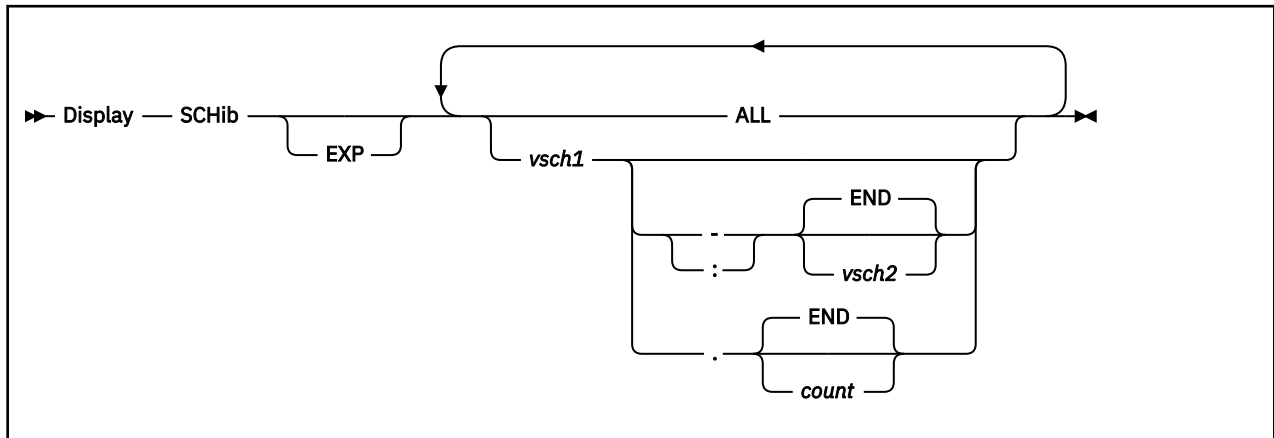
```
BEA = mmmmmmmmmmmmmmm
```

is the response you receive when you display the breaking-event-address register. The contents of the breaking-event-address register are displayed in hexadecimal format.

For ESA/390 and ESA/XC mode guests, the breaking-event-address is displayed with 4 bytes. In z/Architecture mode, the breaking-event-address is displayed with 8 bytes.

In z/Architecture mode the high and low half of the register may be separated by an optional underscore for readability. To have the underscore displayed, use the SET UNDERSCORE command.

## DISPLAY SCHIB



### Authorization

Privilege Class: G

### Purpose

Use DISPLAY SCHIB to see your subchannel information blocks (SCHIB).

### Operands

#### EXP

specifies that extended channel measurement information is to be displayed.

#### ALL

specifies that SCHIBs for all virtual devices are to be displayed.

#### *vsch1*

identifies the starting (or only) virtual subchannel whose SCHIB is to be displayed.

#### : END

#### - *vsch2*

*vsch2* specifies that all SCHIBs in the range *vsch1* to *vsch2* are to be displayed, or that all SCHIBs from *vsch1* to the highest number subchannel are to be displayed. END is the default if you specify a hyphen (-), or a colon (:). A colon (:) means the same as a hyphen (-) when specifying a range.

#### END

#### . *count*

*count* identifies the number of consecutive SCHIBs starting at *vsch1* that are to be displayed. Specify END to display all SCHIBs from *vsch1* to the highest number subchannel. END is the default if you specify a period (.).

### Usage Notes

1. If you specify a list of SCHIBS, and one is not found, the DISPLAY command terminates at the first SCHIB not found. For example, if you enter `display SCHIB 0022 0033 0055 0011`, and subchannel 0055 is not found, only SCHIB 0022 and 0033 will be displayed.

If you specify a range of SCHIBs, the DISPLAY command terminates if the first SCHIB in the range is not found. Otherwise, all those SCHIBs that were found are displayed.

2. You can use the QUERY VIRTUAL command to determine the subchannel number associated with a virtual device.

3. The response to this command is a header line followed by a line of information for each SCHIB that you specify. (The header line identifies, column by column, the SCHIB information below it. See “Responses” below.)
4. For more information, see [“Usage Notes” on page 331](#).

## Responses

### Response 1: When EXP is not specified

```
SCHIB DEV INT-PARM ISC FLG LP PNO (Con't)
nnnn nnnn nnnnnnnn n nn nn nn

LPU PI MBI PO PA CHPID0-3 CHPID4-7
nn nn nnnn nn nn nnnnnnnn nnnnnnnn
```

If an interrupt is pending for the virtual subchannel, the following line describes the pending interrupt:

```
KEY ECC FPI ZN SCTL CCW-ADDR DS SS CNT EXT-LOG
n n n n nnnn nnnnnnnn nn nn nnnn nnnnnnnn
```

### Response 2: When EXP is specified

```
SCHIB DEV INT-PARM ISC FLG MBI FLG2 MBA
nnnn nnnn nnnnnnnn n nn nnnn nn nnnnnnnnnnnnnnnnn
```

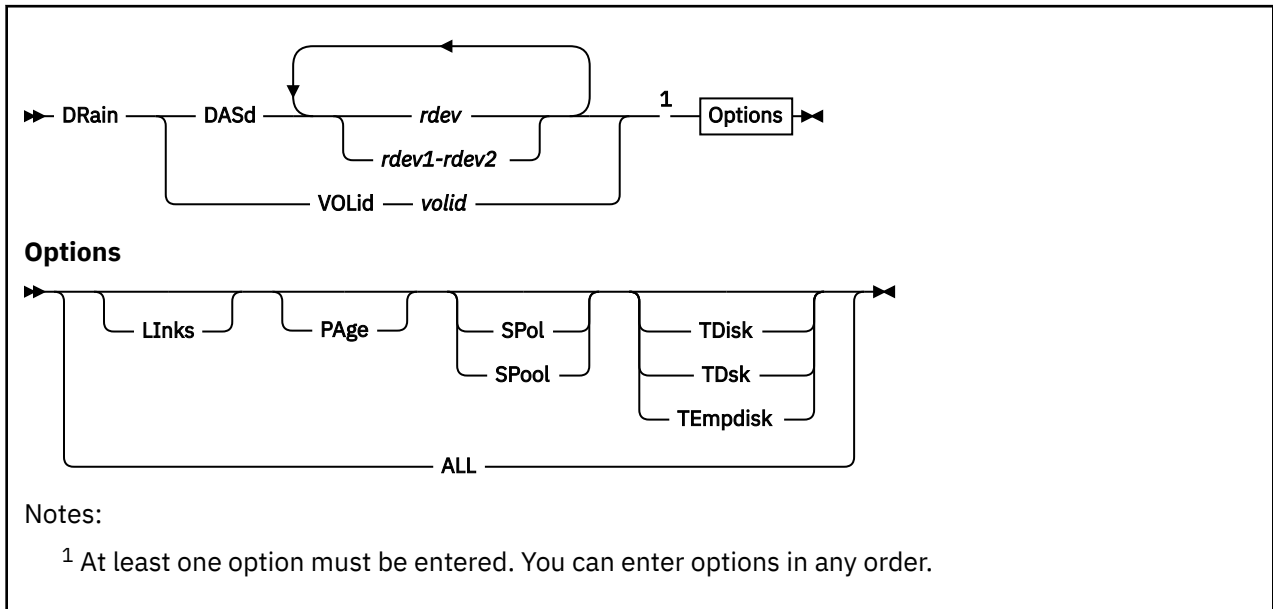
Response header lines contain headings from the list below. A brief description is included.

Heading	Description
INT-PARM	The interrupt parameter
ISC	The interrupt subclass
FLG	The flag field
LP	The logical-path mask
PNO	The path-not-operational mask
LPU	The last-path-used mask
PI	The path-installed mask
MBI	The measurement block index
	<b>Note:</b> This will be ----- when format-1 measurement blocks are being used.
PO	The path-operational mask
PA	The path-available mask
CHPID	The channel path identifiers
FLG2	The flag field (PMCW word 6, bits 24–31)
MBA	The measurement block address
KEY	The SCSW protect key
ECC	The logout condition code bits
FPI	The format, prefetch, and initial response bits
ZN	The initial status condition
SCTL	The function and activity control bytes
CCW-ADDR	The SCSW CCW address
DS	The SCSW device status bits



<b>Heading</b>	<b>Description</b>
SS	The subchannel status bits
CNT	The SCSW residual CCW data count
EXT-LOG	The extended status/time word

## DRAIN (Disk)



### Authorization

Privilege Class: B

### Purpose

Use DRAIN to prevent new allocations on specified real DASDs.

### Operands

#### DASd *rdev*

indicates that you want to drain one or more real DASD. You specify which ones by entering their real device numbers (*rdev*). Each *rdev* must be a hexadecimal device number between X'0000' and X'FFFF'.

#### VOLid *valid*

indicates that you want to drain a real DASD volume, specified by its real volume serial number (*valid*).

### Options

#### ALL

means all of the following:

#### LInks

tells CP not to allow LINK commands to minidisks on this device.

#### PAGe

tells CP not to write pages to this device during page-out operations.

#### SPol

#### SPOOL

tells CP not to allocate space on this device for new spool records.

The hard abend dump spool file or soft abend dump spool file may be open on the device that is being drained. These spool files must be closed for the device to drain completely. Refer to the SET ABEND command and the SET DUMP command for information about closing and reopening these spool files.

For more information about the SET ABEND command, see [“SET ABEND” on page 1413](#). For more information about the SET DUMP command, see [“SET DUMP” on page 1493](#).

**TDisk****TDsk****TEmpdisk**

tells CP not to allocate temporary disk requests to this device.

**Usage Notes**

1. A DASD that has been set to drain a particular usage will appear as if it were full. During allocation attempts, CP continues with the next normal step after encountering a full volume.
2. A DASD set for draining LINKS will not appear mounted when LINK commands are issued.
3. A DASD set for draining may still be attached to a virtual machine.
4. ATTACH and DETACH commands will not affect the various draining status indicators, so you can set the desired draining status before you attach a DASD to the system.
5. Use the QUERY DASD DRAINING command or the QUERY ALLOC command to see the draining status. For more information, see [“QUERY DASD” on page 867](#) and [“QUERY ALLOC” on page 773](#).
6. If you are authorized for some options of this command but not for others and you enter an unauthorized option, your command will be interpreted in the formats for which you are authorized. This may cause the command to run in an unexpected fashion, or reflect an unexpected message.
7. Use the START command to restart a device after it has been drained.

**Responses**

Response 1:

After you enter the DRAIN command, you see the following:

```
Command complete
```

- or -

```
nnnnnnnn device(s) specified; nnnnnnnn device(s)
successfully processed by DRAIN
```

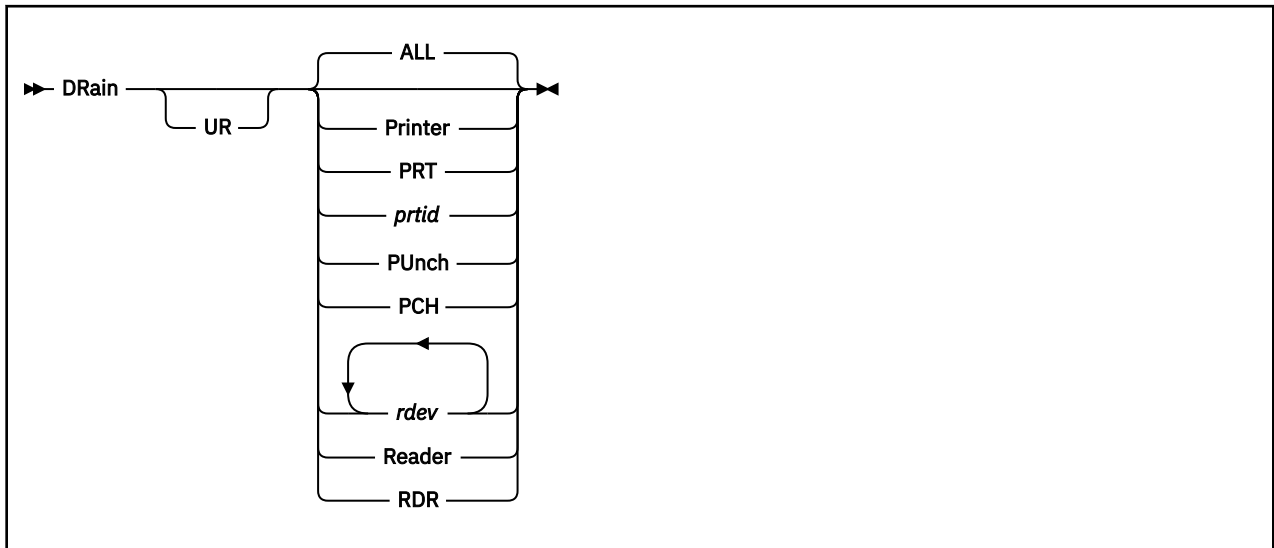
The following response may be sent to the system operator:

```
userid issued DRAIN command for volume volid
on device rdev for alloctype1 [alloctype2] [...]
```

**Messages**

- HCP002E Invalid operand - *operand*
- HCP021E A real device number was not specified or it is invalid.
- HCP023E VOLID missing or invalid
- HCP026E Operand missing or invalid
- HCP040E Device *rdev* does not exist
- HCP009E Invalid range - *range*
- HCP1004E *rdev* is not a DASD volume or is an invalid DASD type.

## DRAIN UR (Unit Record Device)



### Authorization

Privilege Class: D

### Purpose

Use DRAIN (unit record device) to stop spooling operations on specified real unit record devices.

**Note:** Spooling operations stop when the current file finishes processing.

### Operands

#### UR

tells CP you want to drain a real unit record device.

#### ALL

drains all spooled unit record devices in the system. ALL is the default.

#### Printer

#### PRT

#### PUnch

#### PCH

#### Reader

#### RDR

drains all printers, punches, or readers.

#### *rdev*

is a real device number or list of real device numbers of the real spooled devices that you want drained. Each *rdev* must be a hexadecimal device address between X'0000' and X'FFFF'.

#### *prtld*

indicates that the entire command is to be passed for processing to the print server virtual machine designated by *prtld*. The value of *prtld* is provided by the print server virtual machine that will enter the DRAIN command to an AFP. The value is one to eight alphanumeric characters long. If you specify a value of four or fewer characters, *prtld* cannot contain only hexadecimal characters because CP then interprets *prtld* as *rdev*.

## Usage Notes

1. Use the DRAIN command to bring the spooling system or a specified device to a controlled halt. You would normally drain all unit record devices before a system shutdown, and drain a printer before you change the contents of the UCS printer buffer.
2. A device is drained immediately if it is not active when the DRAIN command is issued; otherwise, it is drained when processing of the current file has completed. You can restart a drained device by issuing the START command. If you enter a START command to a device before draining has been completed, the device does not enter drained status, and it continues processing.
3. If you specify more than one real device number, execution of the DRAIN command terminates upon encountering an invalid real device number. None of the remaining devices are drained.
4. For a printer ID, the virtual machine associated with the printer ID processes the DRAIN command. Therefore, the options given on the DRAIN command must be valid for the virtual machine that is doing the processing.

When you enter the DRAIN, DRAIN PRINTER, DRAIN ALL, DRAIN UR, DRAIN UR PRINTER, or DRAIN UR ALL commands, all printers associated with all *prtids* are drained.

## Responses

The following section lists the printer, punch, and reader responses that occur when drained status has been attained. An explanation of the various options that appear in the responses follows the last response.

Inactive Printer Responses:

For every inactive printer that attains the drained status, the first line of the response is the same. The second line varies based upon printer type.

```
PRT rdev DRAINED userid CLASS c ...
PRT rdev FORM form {AUTO } {SEP } {NO3800 } [DEFFCB] LIMIT [NONE]
                  {MANUAL} {NOSEP} {BEG3800} [FILEFCB] [nnnnnnnnnn ]
                  {SETUP } {ANY3800}
```

For **3800 printers** the third line is:

```
PRT rdev FLASH name IMAGE imagelib CHARS name FCB {name} {HOLD } LPP nnn
                                                    {6 } {PURGE}
                                                    {8 }
                                                    {10}
                                                    {12}
```

For **impact printers** the third line is:

```
PRT rdev {NOFOLD} IMAGE imagelib CHARS chars [FCB name] [INDEX nn] LPP nnn
        {FOLD }
```

The fourth line of the response for **all printers** is:

```
PRT rdev DEST {dest1} DEST {dest2 } DEST {dest3 } DEST {dest4 }
              {OFF } { - NONE - } { - NONE - } { - NONE - }
```

The last line of the response for **3800 printers** is:

```
PRT rdev {AFP } {MARK }
        {NOAFP} {NOMARK}
```

The last line of the response for **impact printers** is:

```
PRT rdev {AFP }
        {NOAFP}
```

Inactive Punch Responses:

The following response occurs for each punch that attains the drained status:

```
PUN rdev DRAINED userid CLASS c ...
PUN rdev FORM form {AUTO } {SEP } LIMIT [NONE] (Con't)
                  {MANUAL} {NOSEP} [nnnnnnnnnn ]
                  {SETUP }
```

Continued:

```
PUN rdev DEST {dest1} DEST {dest2 } DEST {dest3 } DEST {dest4 }
              {OFF }      {- NONE -} {- NONE -} {- NONE -}
```

Inactive Reader Response:

```
RDR rdev DRAINED SYSTEM
```

Active Printer Responses:

For every active printer that attains the drained status, the first line of the response is the same. The second line varies according to printer type.

```
PRT rdev DRAINING userid CLASS c ...
PRT rdev FORM form {AUTO } {SEP } {N03800 } [DEFFCB] LIMIT [NONE]
                  {MANUAL} {NOSEP} {BEG3800} [FILEBCB] [nnnnnnnnnn ]
                  {SETUP } {ANY3800}
```

For **3800 printers** the third line is:

```
PRT rdev FLASH name IMAGE imagelib CHARS name FCB {name} {HOLD } LPP nnn
                                                    { }
                                                    6 {PURGE}
                                                    { }
                                                    8 { }
                                                    { }
                                                    10 { }
                                                    { }
                                                    12 { }
```

For **impact printers** the third line is:

```
PRT rdev {NOFOLD} IMAGE imagelib CHARS chars [FCB name] [INDEX nn] LPP nnn
        {FOLD }
```

The fourth line of the response for **all printers** is:

```
PRT rdev DEST {dest1} DEST {dest2 } DEST {dest3 } DEST {dest4 }
              {OFF }      {- NONE -} {- NONE -} {- NONE -}
```

The next line of the response for **3800 printers** is:

```
PRT rdev {AFP } {MARK }
        {NOAFP} {NOMARK}
```

The next line of the response for **impact printers** is:

```
PRT rdev {AFP }
        {NOAFP}
```

The last line of the response for **all printers** is:

```
PRT rdev PRINTING userid FILE spoolid RECS {nnnn} COPY [*]nnn SEQ sss
                                           {nnnK}
                                           {nnnM}
```

Continued:

```
CL c typ
```

You receive a message when the system begins processing a spool file that has been sent to the real device with the MSG attribute specified with the SPOOL command.

**Note:** CP itself does not provide a response for *prtids*. The virtual machine associated with *prtid* provides the response.

Active Punch Responses:

The following response occurs for each punch that attains the drained status:

```
PUN rdev DRAINING userid CLASS c ...
PUN rdev FORM form {AUTO } {SEP } LIMIT [NONE] (Con't)
                  {MANUAL} {NOSEP} [nnnnnnnnnn ]
                  {SETUP }
Continued:
PUN rdev DEST {dest1} DEST {dest2 } DEST {dest3 } DEST {dest4 } (Con't)
              {OFF }      {-NONE -} {- NONE -} {- NONE -}
Continued:
PUN rdev PUNCHING userid FILE spoolid RECS {nnnn} COPY [*]nnn SEQ sss
                                           {nnnK}
                                           {nnnM}
Continued:
CL c typ
```

Active Reader Response:

```
RDR rdev DRAINING SYSTEM
```

Explanation of Responses:

All the various options used in the preceding response section are detailed below.

#### **PRT**

#### **PUN**

#### **RDR**

*rdev* is the device that you drained or are draining.

#### **rdev**

*rdev* is the real device number of the drained or draining device.

#### **CLASS c**

*c* are those output classes that the drained or draining device was servicing. One to eight classes are displayed for each printer or punch device that was drained or is draining.

#### **FORM form**

*form* changes the form name of a spool file. *form* is one to eight characters long.

#### **AUTO**

**AUTO** indicates that the operator wants CP to schedule forms output. The currently active form is processed first. When all files with the current form have been printed or punched, CP automatically selects a new form for the device. The operator receives a message to change forms, and the output device is idle. The operator can then put in the new form and press the START button (or enter the START command) to print files with the new form.

#### **MANUAL**

**MANUAL** indicates that the operator schedules output forms. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is printed or punched, the output device stops. The operator can then schedule a new form by issuing a START command with the FORM option.

#### **SETUP**

**SETUP** indicates that the operator wants CP to print a page so that forms alignment can be verified. (This option is valid only for impact printers.) The operator gets a message indicating that the form should be set up. When the form has been set up, the operator should press the STOP button, then the START button. An alignment page is then printed. It is the first page of the file, with letters replaced with Xs and the numbers replaced with 9s. Each time the operator presses the STOP or START buttons after

a full alignment page is printed, another alignment page is printed. When the forms are aligned, you should enter the START command for the device. The printer then returns to its previous mode (AUTO or MANUAL), and the file prints in its entirety.

If neither MANUAL, AUTO, nor SETUP is specified and the printer is currently in SETUP mode, it returns to the previous mode, AUTO or MANUAL. If no START command has been issued with the MANUAL, AUTO, or SETUP options since a system cold start, the default is MANUAL.

**SEP  
NOSEP**

indicates the current separator status.

**LIMIT NONE****LIMIT *nnnnnnnnnn***

indicates the current value for LIMIT, which was set by the CP START command or the system configuration file.

**NO3800****BEG3800****ANY3800**

specifies the restrictions on the 3800 load CCWs in effect after the device is started again.

**DEFFCB****FILEFCB**

specifies the FCB that is currently in use. DEFFCB means that the default that was loaded in the printer is used. FILEFCB means that file FCBs are sent to the real printer.

***rdev***

is the real 3800 printer or list of 3800 printer numbers that you want drained.

**FLASH *name***

signifies that a form overlay contained in the 3800 printer is to be superimposed (or flashed) onto the specified number of copies of the output. The name, to be one to four characters long, identifies the forms overlay to be used in the projection.

**IMAGE *imagelib***

specifies the name of the system image library file that contains the load images (character sets), FCBs, graphic character modifications, or copy modifications that are to be used for printing files and separators. *imagelib* may be from one to eight characters long, and must be an existing image library file.

**FCB *name***

is the name of the forms control buffer (FCB) module to be used by the printer in the vertical formatting of a page.

**Note:** This does not appear for 1403-type printers.

**CHARS *name***

specifies the name of the default character set used when spool files are being printed that do not specify a character set.

**HOLD**

prevents the printing of a file by placing it in a user hold status.

**PURGE**

closes and purges the spool file on the specified 3800 printer, regardless of the cont, keep, and hold setting for the device.

**LPP *nnn***

indicates the lines per page that the printer is using for the generation of separator pages. The LPP value is determined from the FCB loaded on the printer.

If no FCB has been loaded to the printer, the LPP will appear as zero. Since the FCB is actually loaded when a file is printed, it is possible that the LPP will appear as zero or the LPP value for the prior FCB after the START command with the FCB operand is issued.



**Note:** If the number of lines per page derived from the FCB data is less than 30 or greater than 255, the LPP for that printer will be set to a default value of 66 for impact printers, or 60 for 3800 printers.

**NOFOLD****FOLD**

indicates whether lowercase characters are being translated (folded) into uppercase for printing.

**CHARS *chars***

indicates the character set being used by the printer.

**INDEX *nn***

indicates the character position on the page at which to begin printing.

**PRT****PUN**

indicates the type of device on which the file is being processed.

***rdev***

is the real device number.

**DEST *dest***

is the destination value. *dest* is a one- to eight-character value. OFF is the default destination for the device for the first of four destination values. The designation - NONE - is shown for other destination values that have not been set.

**AFP****NOAFP**

specifies whether the printer is to process files with advanced function printer (AFP) characteristics.

**MARK****NOMARK**

indicates whether 3800 separator trailer pages are being marked with separator bars.

***userid***

is the name of the spool file owner.

**FILE *spoolid***

is the spool file number.

**RECS *nnnn*****RECS *nnnK*****RECS *nnnM***

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

**COPY *nnn*****COPY \* *nnn***

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (\*) is specified, the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

**SEQ *sss***

is the sequence number for the file on the printer or punch.

**CL *c***

is the spool file class.

***typ***

shows the device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP, or SYS.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP031E Buffer missing or invalid; [FCB too long|extended FCB not supported]
- HCP036E Index missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP043E Buffer *buffer* does not exist
- HCP046E *type rdev* offline
- HCP140E *type {rdev/ldev}* attached to *userid*
- HCP148E Printer *rdev* buffer load *name* failed
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP418E {*rdev/vdev*} is not a spooled device.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*
- HCP1552E More than {4|8} operands were specified for the {CHARS|CLASS} option.
- HCP2500E Image library *libname* could not be found [for printer *rdev*].
- HCP2501E A spool paging I/O error occurred while reading image [[library] *imagename* on printer *rdev*
- HCP6005E Option *option* is not supported by z/VM.
- HCP6525E Authorization request failed.

# DUMP

---

## Purpose

Use the DUMP commands to send data to your printer from the following machine components, both real and virtual:

- The contents of first-level storage (host real storage or host logical storage)
- The contents of second-level storage (the storage that appears real to the operating system running in your virtual machine)
- The contents of third-level storage (the storage that appears virtual to the operating system running in your virtual machine)
- Old and new PSWs and interrupt information for the following interrupts:
  - SVC
  - I/O
  - PROGRAM
  - EXTERNAL
  - MACHINE CHECK
- Storage keys
- Linkage stack
- Contents of registers:
  - Access
  - Control
  - Floating-point
  - Floating-point control
  - General purpose
  - Prefix
- Subchannel information blocks (SCHIBs)

## Usage Notes

1. The DUMP command prints the contents of various storage locations, registers, and control registers to your virtual printer. You may be able to detect the cause of program errors by examining this information.
2. Using a single command, you can dump multiple storage locations, registers, and control words.  
Follow these rules to execute the command properly:
  - a. Do not put blanks between operands that relate to a single DUMP request, except for PSW and SCHIB, which require at least one blank between the first and second operands. For example, if you want to dump hexadecimal location 40, enter `dump H40`.
  - b. Separate each set of dump requests from others by at least one blank.  
For example, if you want to dump the hexadecimal location 40 and the contents of register 5, enter `dump H40 G5`.
3. You can assign a *dumpid* to any dump that you create by means of the DUMP command. Note, however, that even though you can use a single DUMP command to make multiple DUMP requests, you can assign only one *dumpid* per command.

## DUMP

The dump identifier is an asterisk (\*) followed by a 1- to 97-character string (truncated to 80 characters for a z/Architecture mode guest) that is to be assigned to the dump. If specified, the dump identifier must be the last operand you enter on the command line. Blanks are permitted in the *dumpid*.

For example, `dump G *general registers` creates a dump of your general registers and assigns the dump the identifier "general registers".

4. As each request is processed, a message is sent to your display device. This permits you to stop DUMP processing by pressing the PA1 key. When the command terminates, you may enter another command.
5. The defaults for the following DUMP operands are:

<b>Operand</b>	<b>Default Before Rounding</b>
G	0-end
G-	0-end
G-5	0-G5
G5-	G5-end
G5	G5
T	T0
T-	T0-end
T-1000	T0-1000
T1000-	T1000-end
T1000	T1000
PSW	current

6. When dumping a range of addresses or subchannels, the command operand must be followed by an address, a range, or a colon (:), hyphen (-), or period (.) which forms a range. When dumping registers, the command operand can stand alone.
7. If you enter a command line that contains an operand that is not valid, the DUMP command terminates when it encounters that operand. Any previous valid operands are processed before termination occurs.
8. The DUMP command lets you dump information in the following formats:
  - Hexadecimal string / word segments
  - EBCDIC translation for uppercase letters and numbers 0 through 9
  - Hexadecimal word segments with EBCDIC and ASCII translation
  - Instruction
  - Storage key

**Note:** The EBCDIC translation is displayed only when the storage location contains the hexadecimal representation contained in a system translation table. This table is can be tailored by the system programmer via the system configuration file. The IBM-supplied table supports translation for the hexadecimal representation of the digits 0 through 9, the uppercase and lowercase letters A through Z, and a number of special characters. In all other cases, a period (.) is dumped.

If you enter multiple operands on a single command line for a storage or register dump, and do not specify the format that you want the DUMP command processor to use, the last format type that you explicitly entered on the command line defines the type of dump you receive. That is, whenever you explicitly enter a dump type, you implicitly define the default for subsequent operands of the current dump function. If you do not enter any dump format type on the command line, the system defaults are used.

For example, `dump 10 20 T40 80 G12 5 L60-100` dumps the following, respectively:

- Hexadecimal location 10 without EBCDIC translation
- Hexadecimal location 20 without EBCDIC translation
- Hexadecimal location 40 with EBCDIC translation
- Hexadecimal location 80 with EBCDIC translation
- The contents of general register 5
- The contents of general register 12
- Hexadecimal locations 60 through 100 without EBCDIC translation

The DUMP command suppresses identical lines and imbeds a response whenever a 64K boundary is crossed.

9. The DUMP storage command supports indirect addressing. Indirect addressing tells the command to use the **contents** of an address you specified instead of the address itself. This can help reduce the number of commands you issue and the tedious tracking of address pointers. You specify indirect addressing with an **&** (for 64-bit) or a **%** (for 31-bit).

The following is an example of double indirection using the % for 31-bit addressing. First, the storage is shown using the usual DUMP without indirect addressing. Following that is the same storage displayed using indirect addressing with only one DUMP command (and without having to know that the information you want is at address 3F1AB40). The information displayed is a LNGBK control block for the system default language using a pointer from the CP nucleus SYSCM area.

```
DU 3EF1030
R03EF1030 0000524C
```

```
DU 524C
R0000524C 03F1AB40
```

```
DU T3F1AB40.80
```

```
R03F1AB40 00000000 00000000 00000000 00000000 06 *.....*
R03F1AB50 to 03F1AB5F suppressed line(s) same as above ...
R03F1AB60 00390000 00000000 C1D4C5D5 C7000000 *.....AMENG...*
R03F1AB70 00000000 03F1A878 00000000 0041E000 *.....1y.....*
R03F1AB80 C8C3D7D4 C5E24040 00000000 00000000 *HCPMES .....*
R03F1AB90 00000000 00000000 00000000 00000000 *.....*
R03F1ABA0 to 03F1ABAF suppressed line(s) same as above ...
R03F1ABB0 4CD3D5C7 4C4C4C4C 80000070 00D5D3E3 *<LNG<<<<.....NLT*
```

```
DU T3EF1030%%.80
```

```
R03F1AB40 00000000 00000000 00000000 00000000 06 *.....*
R03F1AB50 to 03F1AB5F suppressed line(s) same as above ...
R03F1AB60 00390000 00000000 C1D4C5D5 C7000000 *.....AMENG...*
R03F1AB70 00000000 03F1A878 00000000 0041E000 *.....1y.....*
R03F1AB80 C8C3D7D4 C5E24040 00000000 00000000 *HCPMES .....*
R03F1AB90 00000000 00000000 00000000 00000000 *.....*
R03F1ABA0 to 03F1ABAF suppressed line(s) same as above ...
R03F1ABB0 4CD3D5C7 4C4C4C4C 80000070 00D5D3E3 *<LNG<<<<.....NLT*
```

10. If you issue DISPLAY with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the DISPLAY is rejected and error message HCPCLD6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:

- SYSTEM CLEAR
- IPL by NSS name
- IPL by device with the CClear option

- SET MACHine (to a different machine architecture)

## Messages

(General for DUMP)

- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*
- HCP005E Invalid hexdata - *hexdata*
- HCP009E Invalid range - *range*
- HCP010E Invalid register - *register*
- HCP026E Operand missing or invalid
- HCP060E {Dump failed|Printer output terminated}; virtual printer unavailable
- HCP061E Dump failed; virtual printer error
- HCP160E Hexloc *hexloc* exceeds storage
- HCP164E Hexloc *hexloc* non-addressable storage
- HCP260E {Command|*command* command} processing cannot complete due to errors.
- HCP267E A subchannel number was not supplied or is invalid.
- HCP410E CP ENTERED; PAGING ERROR
- HCP422E The same option was specified twice.
- HCP1001E An operand is missing for *option*.
- HCP1003E {*CP module name|rdev|ldev*} {is|was} not found.
- HCP1459E The virtual machine is placed in check-stop or system check-stop state due to a system malfunction with CPU *nn*.
- HCP6150E An option has been specified that is not valid in {ESA/390|ESA/XC|z/Architecture|z/XC} mode.
- HCP6151E Storage {operand} is not addressable because of { a protection | an addressing | a segment translation | a page translation | a translation specification | an ASN translation | an AFX translation | an ASX translation | an ALET translation | an ALEN translation | an ALE sequence | an ASTE validity | an ASTE sequence } exception.
- HCP6155E Storage is not addressable because of an invalid PSW.
- HCP6156E {ASIT *asit*|Address space identifier *spaceid*} does not designate an address space to which you have the appropriate access.
- HCP6578E Invalid {access register|ASIT|address space identifier} - *option*
- HCP6525E Authorization request failed.

## DUMP ACCESS LIST

```

▶▶ DUmP — ACCList 1 —————▶▶
                    |
                    | *dumpid
                    |

```

Notes:

<sup>1</sup> Do not put blanks between operands that relate to a single DUMP request.

### Authorization

Privilege Class: G

### Purpose

Use DUMP ACCESS LIST to dump entries in your virtual machine's host access list.

### Operands

#### *\*dumpid*

is an asterisk followed by a 1- to 97-character identifier that is to be assigned to the dump. If specified, *dumpid* must be the last operand you enter on the command line.

### Usage Notes

1. If you write a program or EXEC that parses the output of this command, you should avoid a dependency on the ordering of the columns in the output. Your program or EXEC should use the heading line to determine which column contains a particular piece of information.
2. For more information, see [“Usage Notes” on page 395](#).

### Responses

Response 1:

```

ALET      ACC   FAULTS  ASIT      SPACE IDENTIFICATION  (Con't)
tttttttt {R/O} {ASYNCH} {aaaaaaaaaaaaaaaa}  sssssssssssssssssssssssssssssssssss
          {R/W} {SYNCH } {**REVOKED** }

```

Continued:

```

nnnn TOTAL ENTRIES:  vvvv VALID,  rrrr REVOKED,  uuuu UNUSED

```

#### *ttttttt*

is the access list entry token (ALET) corresponding to the host access-list entry (8 hexadecimal digits).

#### **R/O**

indicates that the host access-list entry can be used only for read references to the address space.

#### **R/W**

indicates that the host access-list entry can be used for read and write references to the address space.

#### **ASYNCH**

indicates that page-fault handshaking applies to the host access-list entry.

#### **SYNCH**

indicates that page-fault handshaking does not apply to the host access-list entry.

## DUMP ACCESS LIST

### **aaa...aaa**

is the address space identification token (ASIT) for the address space designated by the host access-list entry (16 hexadecimal digits).

### **\*\*REVOKED\*\***

indicates that the host access-list entry previously designated an address space, but access to that address space has been revoked (the host access-list entry is in the revoked state). The *spaceid* field is left empty for revoked host access-list entries.

### **sss...sss**

is the *spaceid* for the address space designated by the host access-list entry (3 to 33 characters).

### **nnnn TOTAL ENTRIES**

is the total number of entries in the host access list.

### **vvvv VALID**

is the number of entries in the host access list that designate address spaces, and can be used to address data in those address spaces (host access-list entries that are in the valid state).

### **rrrr REVOKED**

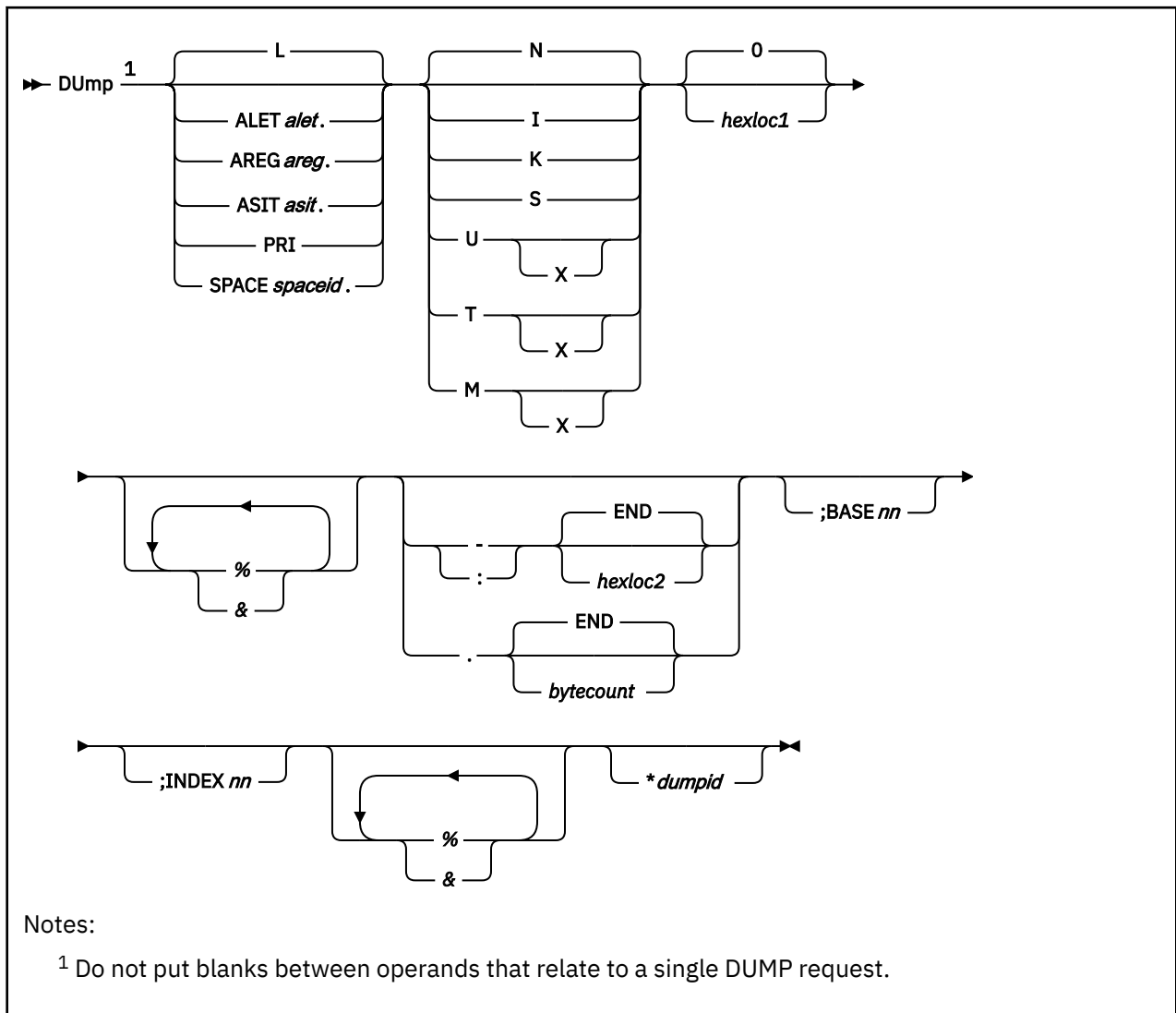
is the number of entries in the host access list that previously designated address spaces, but which can no longer be used to address data because the program's privileges to do so have been withdrawn.

### **uuuu UNUSED**

is the number of entries in the host access list that remain available for use (host access-list entries that are in the unused state).



## DUMP (Guest Storage - ESA/XC, z/XC)



### Authorization

Privilege Class: G

### Purpose

Use DUMP (Guest Storage - ESA/XC, z/XC) to dump the storage contents of an ESA/XC or z/XC virtual machine.

### Operands

**L**

dumps the contents of storage based on the setting of the address-space-control field in the guest PSW. This is the default.

If the virtual CPU is in the primary-space mode, the contents of real storage in the host-primary address space are dumped. If the virtual CPU is in the access-register mode and a nonzero BASE*nn* operand is also specified, the contents of real storage in an access register specified address space are dumped.

## DUMP (Guest Storage - ESA/XC)

If the virtual CPU is in the access-register mode, but either the *BASEnn* operand is not specified or is specified as 'BASE0', the contents of real storage in the host-primary address space are dumped.

### **ALE***alet.*

umps the contents of real storage in an access list entry token (ALET) specified address space. The *alet* may be 1 to 8 hexadecimal digits; leading zeros are optional. The ALET is used in conjunction with the virtual machine's host access list to determine the address space to dump.

### **AREG***areg.*

umps the contents of real storage in an access register specified address space. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The ALET within the access register is used in conjunction with the virtual machine's host access list to determine the address space to dump.

### **ASIT***asit.*

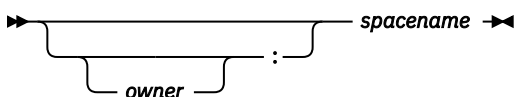
umps the contents of absolute storage in the address space identified by the address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

### **PRI**

umps the contents of real storage in the host-primary address space. This is the address space from which the virtual CPU is fetching instructions at the time the command is run.

### **SPACE***spaceid.*

umps the contents of absolute storage in the address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the address space, and *spacename* is the 1-character to 24-character address-space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

### **N**

umps the contents of storage in hexadecimal without character translation. This is the default. Storage is dumped in word segments. An address range is dumped starting at the fullword in which the specified address resides. Storage is dumped in fullword increments.

The storage key is dumped on the first line, as well as at every page boundary.

### **I**

umps the contents of storage in instruction format. The specified address is rounded down to the nearest halfword boundary.

The storage key is dumped on the first line, as well as at every page boundary.

### **K**

umps the storage keys in hexadecimal for all page frames or partial page frames included in the specified address range.

### **S**

umps the contents of storage in hexadecimal translation. Storage is dumped as a string starting at the location specified. The specified address range is not boundary-aligned.

The storage key is dumped on the first line and at every page boundary.

### **T**

umps the contents of storage in hexadecimal and character form. Storage is dumped in word segments. An address range is dumped starting at the X'10'-byte boundary in which the specified address resides. Storage is dumped in X'10' increments. The hexadecimal data is followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

The storage key is dumped on the first line and at every page boundary.

**U**

umps the contents of storage in character form. Storage is dumped starting at the specified address as an EBCDIC string, or as an ASCII string if the X option is specified. The specified address range is not boundary-aligned.

The storage key is dumped on the first line and at every page boundary.

**M**

umps the contents of storage in hexadecimal and character form. A header is dumped as the first line, and at every page boundary and contains the high-order 32 bits of the 64-bit address along with the storage key. Storage is dumped in word segments. An address range is dumped starting at the X'10' byte boundary in which the specified address resides. The hexadecimal output is followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

**0****hexloc1**

specifies the starting storage location that is to be dumped. The value *hexloc1* may be 1 to 16 hexadecimal digits, leading zeros are optional. The default is zero.

If neither the BASE or INDEX operand is used, *hexloc1* is the address of the first location to be dumped. This address is treated as a 64-bit address, regardless of the addressing mode of the virtual CPU.

If either the BASE or the INDEX operand is used, *hexloc1* is a displacement rather than an address. The first location to be dumped is specified by the sum of *hexloc1* and the contents of the registers specified with the BASE operand or the INDEX operand or both. If general register 0 is specified in the BASE or INDEX operand, it is always treated as containing zero when forming this sum. The sum is treated as a 24-bit, 31-bit, or 64-bit address according to the addressing mode of the virtual CPU.

Storage locations and the minimum amount dumped are rounded according to the following conventions (in hexadecimal):

Option	Start	Minimum Length
T	10	10
M	10	10
K	800	800
S	1	1
I	2	2 - 6 depending on the instruction
N	4	4
U	1	1

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the DUMP command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the DUMP command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**-hexloc2**  
**:hexloc2**  
**-END**  
**:END**

is the last of a range of hexadecimal storage locations whose contents are to be dumped. If specified, *hexloc2* must be equal to or greater than *hexloc1* and must be within the bounds of the address space to be dumped. The address *hexloc2* may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. A colon (:) means the same as a dash (-) when you are specifying a range of addresses.

If neither the BASE or INDEX operand is used, *hexloc2* is the address of the last location to be dumped. This address is treated as a 64-bit address, regardless of the addressing mode of the virtual CPU.

If either the BASE or the INDEX operand is used, *hexloc2* is a displacement rather than an address. The last location to be dumped is specified by the sum of *hexloc2* and the contents of the registers specified with the BASE and INDEX operands. If general register 0 is specified in the BASE or INDEX operand, it is always treated as containing zero when this sum is being formed. The sum is treated as a 24-bit, 31-bit, or 64-bit address according to the addressing mode of the virtual CPU.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify a dash (-) or a colon (:) without an address.

**.bytecount**  
**.END**

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at *hexloc1*) to be dumped. If specified, the *bytecount* must have a value of at least 1 and may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify a period (.) without a byte count value.

**BASEnn**

.dumps the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified general register. The specified register (*nn*) may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

If the virtual CPU is in access-register mode, and the L operand is specified (or defaulted), *nn* designates an access register/general register pair. Access register *nn* is implicitly used, in the same manner as if the AREG operand had been specified, in conjunction with the virtual machine's host access list to determine the address space to dump. If the virtual CPU is not in the access-register mode, or if one of the other space-designation operands (PRI, AREG, ALET, SPACE, ASIT) is specified, access register *nn* is not implicitly used.

**INDEXnn**

.dumps the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

**\*dumpid**

is an asterisk followed by a 1-character to 97-character identifier that is to be assigned to the dump. If specified, the *dumpid* must be the last operand you enter on the command line.

## Usage Notes

1. Special authorization is required to use this command to dump any address space other than the host-primary address space of the virtual CPU. In order to use this command to dump storage in an address space, one of the following conditions must be satisfied:
  - Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.

- If the address space is owned by another virtual machine, your virtual machine must have read-only or read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.

Authorization is granted on a space-by-space basis, so that you may be authorized to dump one address space of a virtual machine but may not be authorized to dump a different space.

2. In addition to dumping the contents of the specified storage, the DUMP (Guest Storage) command dumps the program status words (PSWs), the contents of registers (general, access, floating-point, control, and prefix), and the subchannel information blocks (SCHIBs).
3. When running with a saved segment, you can dump storage locations outside the range of your virtual machine size if they are within the saved segment. If locations exist between the upper limit of your virtual machine and the address at which the saved segment was saved, any attempt to dump those locations (or associated keys) causes the non-addressable storage response to be dumped.
4. The operands L, PRI AREG, and ALET use the host-access-register translation (ART) process. The ART process performs special-case translation processes for certain ALET values and access registers:
  - The ALET X'00000000' always indicates the host-primary address space.
  - Access register 0 is always treated as containing ALET X'00000000'. The contents of access register 0 are not used during host-access-register translation processing.
5. General register 0 is always treated as containing the value 0 when used as the base or index register using the BASE or INDEX operands. The contents of general register 0 are never used.
6. In the XC environment, special rules govern the conversion of real addresses into absolute addresses. In some cases, prefixing is performed as usual to convert a real address to an absolute address, while in other cases prefixing is not performed and real addresses are treated unchanged as absolute addresses.

These rules have an effect on the interpretation of the addresses that specify the range of storage locations to be dumped:

- When storage is specified using the ALET or AREG operand and the ALET used is not X'00000000', prefixing is not performed on the addresses that specify the range of storage in the target address space.
  - When storage in the host-primary address space is specified by using the PRI operand, or by using the ALET or AREG operand when the ALET used is X'00000000', prefixing is performed as usual on the addresses that specify the range of storage in the host-primary address space.
7. Although the R operand is not shown in the syntax for this command, it is accepted for compatibility with the DUMP (Guest Storage) command used for XA and ESA virtual machines. The R operand perform the same function as the L operand and dumps storage from the host-primary address space.
  8. The BASE operand is intended to be used in conjunction with the L operand. It lets you dump the storage used by the virtual machine in one less step than would otherwise be possible. Without the BASE operand, you may need to dump the contents of a general register, then add the displacement, and then enter the DUMP command to examine storage the program is using. With the use of access registers, the BASE option additionally lets you examine storage without needing to know whether the program is currently in the access-register mode. This is done automatically for you by CP.
  9. When multiple DUMP requests are entered on a single line, the BASE and INDEX operands are not applied to subsequent DUMP requests on the command line. They must be explicitly entered for each DUMP request.
  10. During DUMP command processing, key-controlled protection is not used. However, all of the host-related protection mechanisms, such as those applied during host-access-register translation, are used as normal.
  11. If the AREG or ALET operand is specified, the DUMP command uses the host-access-register translation (ART) process to determine the address space to display. This translation process may fail, because of one of several program exception conditions. If the translation does not succeed, the current request is rejected and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the exception condition.

12. When the ASIT or SPACE operand is specified, the DUMP request fails if the specified address space does not exist, or if you are not authorized to access the address space. The current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
13. If you use an underscore, it must be followed by 8 hexadecimal digits.
14. For more information, see [“Usage Notes” on page 395](#).

## Responses

**Note:** The value *hexloc* is a hexadecimal storage address.

Response 1:

```
Rhexloc word1 word2 word3 word4 [key] [*TRANSLATION*]
```

is the response you receive for the T or N option. The EBCDIC translation is given only for the T option. The storage key is dumped on the first line of a request, and at every page boundary.

Response 2:

```
Rhexlochi_ Storage Key - [key]  
hexloclo word1 word2 word3 word4 [*TRANSLATION*]
```

is the response you receive for the M option. The translation is displayed in either EBCDIC or (if the MX option is specified) ASCII. The storage key is displayed on the first line, and at every page boundary.

Response 3:

```
Rhexloc *TRANSLATION* [key]
```

is the response you receive for the U option. The value *hexloc* is the location of the first data byte to be dumped. A minimum of 1 character and a maximum of 32 characters are dumped on each line. The contents of storage are dumped only in translated form and are either EBCDIC or (if the UX option is specified) ASCII. The contents of storage are dumped only in an EBCDIC translation. Periods in your dump represent unprintable characters. The storage key is dumped on the first line of a request and at every page boundary.

Response 4:

```
Rhexloc mnemonic1 x-x ... mnemonic5 x-x [key]
```

is the response you receive for the I option. The value *hexloc* is the location of the first instruction to be dumped. The value *mnemonic* is a character representation of the instruction code; this is a 7-character field in which the mnemonic is left-justified. The expression *x-x* is the actual hexadecimal contents of storage and is 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. A minimum of 1 instruction and a maximum of 5 instructions are dumped on each line. The storage key is dumped on the first line of a request and at every page boundary.

Response 5:

```
Rhexloc stringhexadecimaldata [key]
```

is the response you receive for the S option. The storage key is dumped on the first line of a request and at every page boundary.

Response 6:

```
Rhexloc1 TO hexloc2 KEY = key
```

is the response you receive for the K option. The values *hexloc1* and *hexloc2* are storage locations, and *key* is the associated storage key for *hexloc1*. A response is sent for every contiguous address range that has unique storage keys and whenever a segment boundary is crossed.

## Response 7:

The PSW is used to resolve a logical address. An invalid PSW causes the current request to be rejected and processing to continue with the next request (if any) on the command line. In all cases, a reset PSW (all-zero) is accepted. The rejected operands are displayed along with a response indicating the PSW is invalid. The response is:

```
addr is not addressable because of an invalid PSW.
```

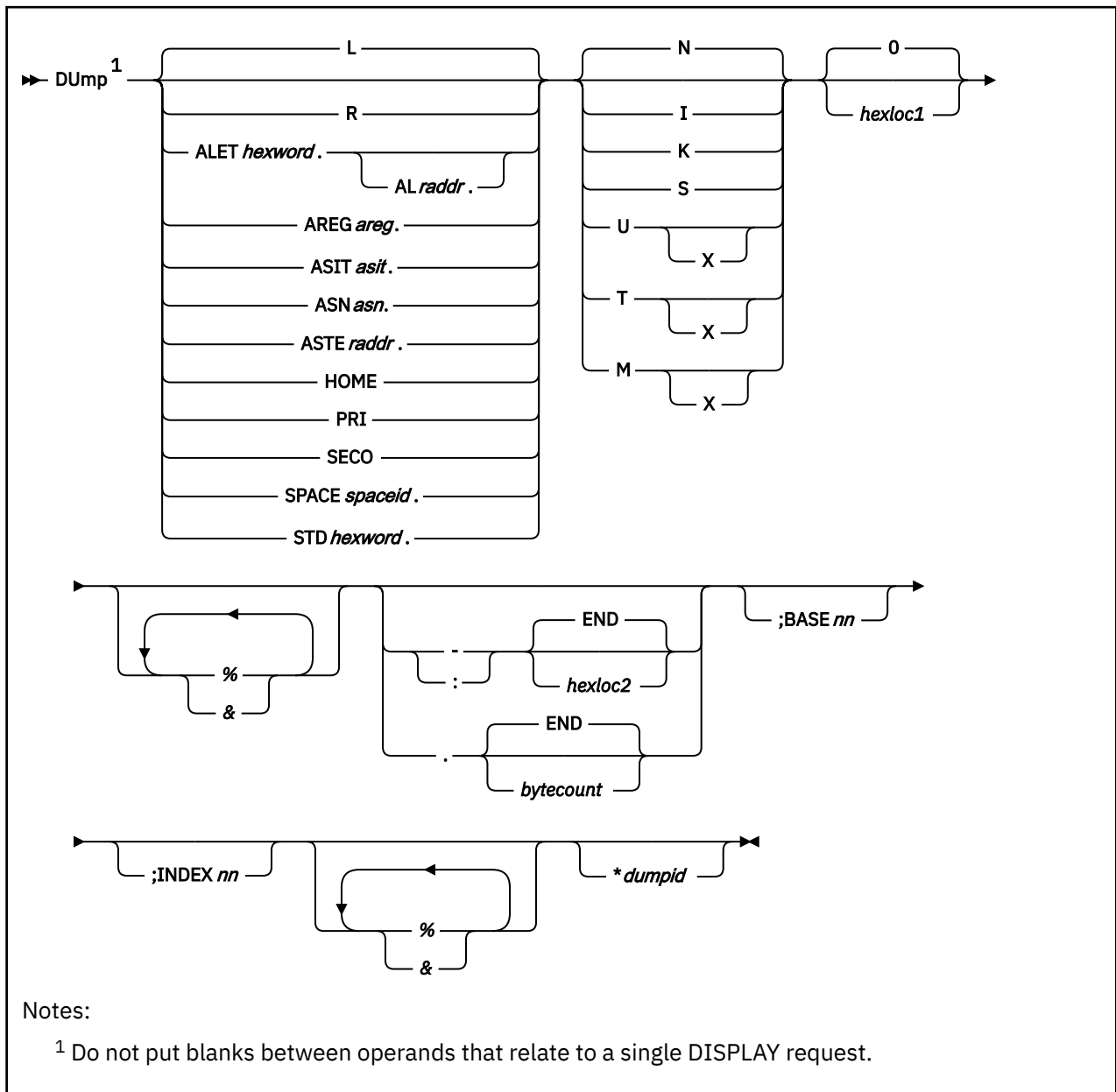
where *addr* is the rejected operand or operands; it specifies an address space and a storage location within that address space.

## Response 8:

If your dump request includes a request to dump identical lines, you receive the following response:

```
Rhexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ....
```

## DUMP (Guest Storage - ESA/390)



### Authorization

Privilege Class: G

### Purpose

Use DUMP (Guest Storage - ESA/390) to dump the storage contents of an ESA/390 virtual machine.

### Operands

**L**

dumps the contents of second-level storage (storage that appears real to your virtual machine) if the virtual machine is operating in DAT OFF mode, or dumps the contents of third-level storage (storage



that appears virtual to your virtual machine) if the virtual machine is operating in DAT ON mode. This is the default.

The address space selected for dumping third-level storage corresponds with the setting of the address space control field in the guest PSW and is the primary, secondary, or (if the virtual machine is an ESA virtual machine) home space. When the PSW indicates access register mode, the primary space is selected for dumping unless a nonzero *BASEnn* operand is also specified and the virtual machine is an ESA virtual machine, in which case an access register specified address space is dumped.

## R

.dumps the contents of second-level storage (storage that appears real to your virtual machine).

### **ALE***Thexword*.

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified access list entry token (ALET). The ALET is used in conjunction with guest access register translation tables to locate the guest address space.

### **AL***raddr*.

specifies an access list (AL) to be used in conjunction with the ALET to locate the guest address space. The guest real address of the access list may be specified as 1 to 8 hexadecimal digits.

### **AREG***areg*.

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the contents of the specified access register. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The ALET within the access register is used in conjunction with guest access register translation tables to locate the guest address space.

### **ASIT***asit*.

dumps the contents of absolute storage in the CP-created address space identified by the specified address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

### **ASN***asn*.

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified address space number (ASN). The *asn* may be 1 to 4 hexadecimal digits. It is used in conjunction with guest ASN-translation tables to locate the guest address space to display.

### **ASTE***raddr*.

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the segment table designation contained within the specified ASN second table entry (ASTE) located at guest real address *raddr*.

## HOME

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 13.

## PRI

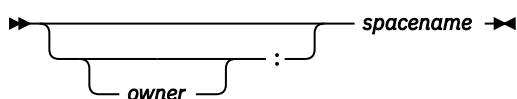
dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 1.

## SECO

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 7.

### **SPACE***spaceid*.

dumps the contents of absolute storage in the CP-created address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the CP-created address space, and *spacename* is the 1-character to 24-character address-space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

**STDhexword.**

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified guest segment table designation.

**N**

dumps the contents of storage in hexadecimal without character translation. This is the default. Storage is dumped in word segments. An address range is dumped starting at the fullword in which the specified address resides. Storage is dumped in fullword increments.

The storage key is dumped on the first line and at every page boundary.

**I**

dumps the contents of storage in instruction format. The specified address is rounded down to the nearest halfword boundary.

The storage key is dumped on the first line and at every page boundary.

**K**

dumps the storage keys in hexadecimal for all page frames or partial page frames included in the specified address range.

**S**

dumps the contents of storage in hexadecimal translation. Storage is dumped as a string starting at the location specified. The specified address range is not boundary-aligned.

The storage key is dumped on the first line and at every page boundary.

**T**

dumps the contents of storage in hexadecimal and character form. Storage is dumped in word segments. An address range is dumped starting at the X'10'-byte boundary in which the specified address resides. Storage is dumped in X'10' increments. The hexadecimal output is followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

The storage key is dumped on the first line and at every page boundary.

**U**

dumps the contents of storage in character form. Storage is dumped starting at the specified address as an EBCDIC string or as an ASCII string if the X option is specified. The specified address range is not boundary-aligned.

The storage key is dumped on the first line and at every page boundary.

**M**

dumps the contents of storage in hexadecimal and character form. A header is dumped as the first line, and at every page boundary and contains the high-order 32 bits of the 64-bit address along with the storage key. Storage is dumped in word segments. An address range is dumped starting at the X'10' byte boundary in which the specified address resides. The hexadecimal output is followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

**O****hexloc1**

is the starting storage location that is to be dumped to your virtual printer. The address *hexloc1* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The default is zero. The starting address will be translated to a X'10'-byte boundary.

Storage locations and the minimum amount dumped are rounded according to the following conventions in hexadecimal:

Option	Start	Minimum Length
T	10	10

Option	Start	Minimum Length
M	10	10
K	800	800
S	1	1
I	2	2 - 6 depending on the instruction
N	4	4
U	1	1

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the DUMP command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the DUMP command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**-hexloc2****:hexloc2****-END****:END**

is the last of a range of hexadecimal storage locations whose contents are to be dumped to your virtual printer. If specified, *hexloc2* must be equal to or greater than *hexloc1* and must be within the virtual machine storage size. The address *hexloc2* may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. A colon (:) means the same as a dash (-) when you are specifying a range of addresses.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify a dash (-) or a colon (:) without an address.

**.bytecount****.END**

is a hexadecimal number designating the number of bytes of storage (starting with the byte at *hexloc1*) to be dumped to your virtual printer. If specified, the *bytecount* must have a value of at least 1 and may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify a period (.) without a byte count value.

**BASEnn**

.dumps the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified general register (*nn*). The general register may be specified as a decimal number from 0 to 15 or as a hexadecimal number from 0 to F. *nn* specifies an AR/GR pair if all of the following are true:

- The L operand is specified.
- The virtual machine is an ESA virtual machine in ESA/390 mode (the default mode).
- The virtual CPU is in access register mode.

The specified access register is used, in the same manner as if the *AREGareg.* operand had been specified, in conjunction with guest access-register-translation tables to locate the guest address space.

**INDEXnn**

dumps the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

**\*dumpid**

is an asterisk followed by a 1-character to 97-character identifier that is to be assigned to the dump. If specified, *dumpid* must be the last operand you enter on the command line.

**Usage Notes**

1. The DUMP (Guest Storage) command dumps the program status words (PSWs), and the contents of registers (general, floating-point, control, and prefix), along with the contents of the specified storage. If your virtual machine is an ESA virtual machine, the subchannel information blocks (SCHIBs) and the access registers are dumped.
2. You can dump storage locations outside the range of your virtual machine size if they are within a saved segment. Any attempt to dump locations (or associated keys) between the upper limit of your virtual machine and the address at which the saved segment is saved causes the non-addressable storage response to appear in the printed output.
3. The operands HOME, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* are valid only if the virtual machine is an ESA virtual machine.
4. When you use the operand *ALETheyword.ALraddr.*, the primary list bit in the ALET is ignored.
5. The operands L, PRI, SECO, HOME, *ASTERaddr.*, *STDhexword.*, *ASNasn.*, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* use the access-register-translation (ART), the ASN translation and the dynamic-address-translation (DAT) processes. These contain special case translation processes:
  - ALET 0 always indicates the current primary address space designated by the STD in guest control register 1, regardless of the current contents of the dispatchable unit access list.
  - ALET 1 always indicates the current secondary address space designated by the STD in guest control register 7, regardless of the current contents of the dispatchable unit access list.
  - Access register 0 is always treated as containing ALET 0. The contents of access register 0 are not used during access register translation processing.
  - General register 0 is always treated as containing the value 0 when used as the base or index register. The contents of general register 0 are never used.
6. The BASE operand is primarily intended to be used in conjunction with the L operand. It lets you dump the storage used by the virtual machine in one less step than would otherwise be possible. Without the BASE operand, you may need to display the contents of a general register, add the displacement, and then enter the DUMP command to examine storage the program is using. With the use of access registers, the BASE option additionally lets you dump storage without needing to know whether your program is currently in access register mode. This is done automatically for you by CP.
7. If you use the *BASEnn* and *INDEXnn* operands together, the specified starting address for a dump is used as a displacement from the address formed from the sum of the contents of the two specified registers.

The values in the registers specified by the *BASEnn* and *INDEXnn* operands are treated as 24-bit or 31-bit real addresses according to the addressing mode (A) bit in the virtual machine's PSW. That is, if the A bit is 1, the value is a 31-bit address. If the A bit is 0, the value is a 24-bit address.

8. During DUMP command processing, the authorization mechanisms used during normal address translation are not used.
9. Special authorization is required to use the SPACE or ASIT operand of this command to dump any address space other the host-primary address space of the virtual CPU. In order to use this command to dump storage in an address space, one of the following conditions must be satisfied:

- Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.
- If the address space is owned by another virtual machine, your virtual machine must have read-only or read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.

Authorization is granted on a space-by-space basis, so that you may be authorized to dump one address space of a virtual machine but may not be authorized to view a different space.

10. The operands V and W used in dumping third-level storage from guest primary and secondary storage are supported for compatibility. The PRI and SECO operands replace the V and W operands.
11. When the ASIT or SPACE operand is specified, the DUMP request fails if the specified address space does not exist, or if you are not authorized to access the address space. The current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
12. The DUMP command requires a segment table designation (STD) to display third-level storage (storage that appears virtual to your virtual machine). It may receive the STD as an operand, find it in a control register, or use the access-register translation (ART) or the address space number (ASN) translation process to obtain the STD. The access-register translation and ASN translation processes may fail because of one of several program exception conditions. If the translation does not succeed, the current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a response indicating the nature of the exception condition.
13. When multiple DUMP requests are entered on a single line, the BASE $nn$  and INDEX $nn$  operands are not applied to subsequent DUMP requests on the command line. They must be explicitly entered for each DUMP request.
14. If you use an underscore, it must be followed by 8 hexadecimal digits.
15. If you issue DUMP with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the DUMP command is rejected and error message HCPCDL6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CClear option
  - SET MACHine (to a different machine architecture)
16. For more information, see [“Usage Notes” on page 395](#).

## Responses

### Notes:

1. The value *hexloc* is a hexadecimal storage address.
2. The character displayed for the variable *t* in the string *thexloc* indicates the type of storage:
  - R indicates real storage (second-level storage).
  - V indicates virtual storage (third-level storage).
3. In responses that include third-level storage, an *Rhexloc* trailer shows the second-level address that corresponds to the third-level address. The trailer is displayed on the first line of a range and at every page boundary.
4. Responses 10 and 11 appear on your display screen.

Response 1:

```
thexloc word1 word2... word8 [key] [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the T or N option. The translation is given only for the T option and is in either EBCDIC or (if the TX option is specified) ASCII. The storage key is dumped on the first line and at every page boundary.

Response 2:

```
thexlochi_ Storage Key - [key]
hexloclo word1 word2 word3 word4 [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the M option. The translation is displayed in either EBCDIC or (if the MX option is specified) ASCII. The storage key is displayed on the first line, and at every page boundary.

Response 3:

```
thexloc *TRANSLATION* [key] [Rhexloc]
```

is the response you receive for the U option. The value *hexloc* is the location of the first data byte to be dumped. A minimum of 1 character and a maximum of 64 characters per line are displayed. The contents of storage are dumped in translated form and are in either EBCDIC or (if the UX option is specified) ASCII. Periods in your dump represent unprintable characters. The storage key is dumped on the first line of a request and at every page boundary.

Response 4:

```
thexloc mnemonic1 x-x ... mnemonic5 x-x [key] [Rhexloc]
```

is the response you receive for the I option. The value *hexloc* is the location of the first instruction to be dumped. The value *mnemonic* is a character representation of the instruction code; this is a 7-character field in which the mnemonic is left-justified. The expression *x-x* is the actual hexadecimal contents of storage and is 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. A minimum of 1 instruction and a maximum of 5 instructions are dumped per line. The storage key is displayed on the first line of a request and at every page boundary.

Response 5:

```
thexloc stringhexadecimaldata [key] [Rhexloc]
```

is the response you receive for the S option. The storage key is displayed on the first line of a request and at every page boundary.

Response 6:

```
thexloc1 TO hexloc2 KEY = key
```

is the response you receive for the K option. The values *hexloc1* and *hexloc2* are storage locations, and *key* is the associated storage key for *hexloc1*. A response is dumped for every contiguous address range that has unique storage keys, and whenever a segment boundary is crossed.

Response 7:

The PSW is used to resolve a logical address. A PSW that is not valid causes the current request to be rejected, and processing continues with the next request (if any) on the command line. In all cases, a reset PSW (all-zero) is accepted. The rejected operands are displayed along with a response indicating that the PSW is not valid. The response is:

```
addr is not addressable because of an invalid PSW.
```

where *addr* is the rejected operand or operands; it specifies an address space and a storage location within that address space.

Response 8:

The dynamic address translation (DAT) process uses the segment table designator (STD) to convert a third-level address to a second-level address (storage that appears real to your virtual machine). The translation process may generate one of several exceptions, identifying why one or more pages of storage are not addressable. A range of rejected storage addresses are displayed along with a response indicating the nature of the exception condition. The responses are:

```

thexloc1 to hexloc2 nonaddressable storage
- protection exception
thexloc1 to hexloc2 nonaddressable storage
- addressing exception
thexloc1 to hexloc2 nonaddressable storage
- segment translation exception
thexloc1 to hexloc2 nonaddressable storage
- page translation exception
thexloc1 to hexloc2 nonaddressable storage
- translation specification exception

```

where *hexloc1* is the starting address and *hexloc2* the ending address of the non-addressable storage. The addresses take into account basing, indexing, and wrapping and are padded to 8 digits.

Response 9:

If your request includes a request to dump identical lines, you receive the following response:

```

thexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...

```

Response 10:

```

DUMPING LOC thexloc

```

is sent to your console to indicate that a segment boundary has been reached. Each segment boundary location has an address that ends in X'00000'. The value *hexloc* identifies the storage boundary reached.

Response 11:

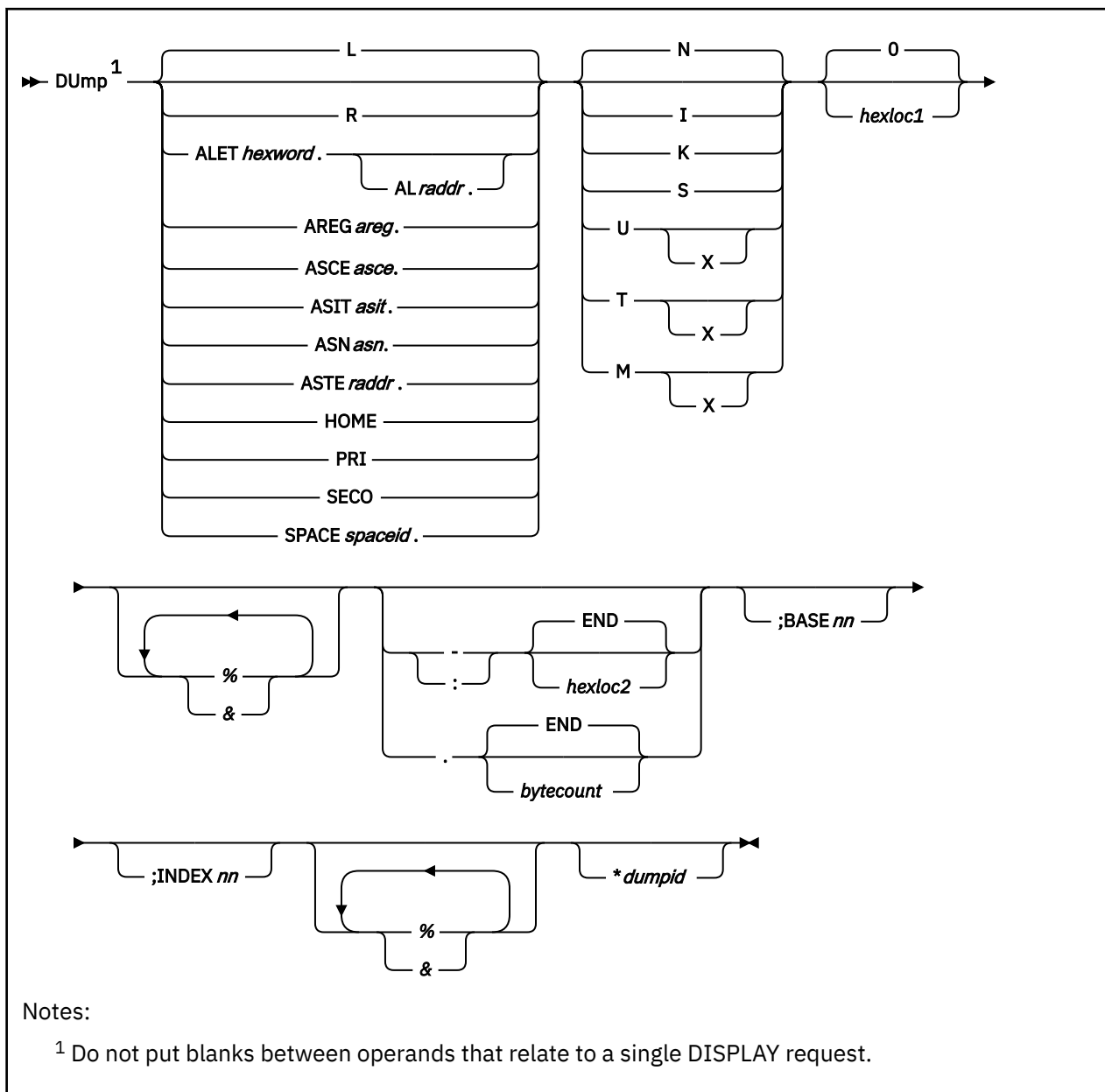
```

Command complete

```

is issued when the command completes successfully.

## DUMP (Guest Storage - z/Architecture)



### Authorization

Privilege Class: G

### Purpose

Use DUMP (Guest Storage - z/Architecture) to dump the storage contents of a virtual machine in z/Architecture mode.

### Operands

**L**

dumps the contents of second-level storage (storage that appears real to your virtual machine) if the virtual machine is operating in DAT OFF mode, or dumps the contents of third-level storage (storage



that appears virtual to your virtual machine) if the virtual machine is operating in DAT ON mode. This is the default.

The address space selected for dumping third-level storage corresponds with the setting of the address space control field in the guest PSW and is the primary, secondary, or (if the virtual machine is an ESA virtual machine) home space. When the PSW indicates access register mode, the primary space is selected for dumping unless a nonzero *BASEnn* operand is also specified and the virtual machine is an ESA virtual machine, in which case an access register-specified address space is dumped.

## R

.dumps the contents of second-level storage (storage that appears real to your virtual machine).

### **ALETheyword.**

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified access list entry token (ALET). The ALET is used in conjunction with guest access register translation tables to locate the guest address space.

### **ALraddr.**

specifies an access list (AL) to be used in conjunction with the ALET to locate the guest address space. The guest real address of the access list may be 1 to 8 hexadecimal digits.

### **AREGareg.**

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the contents of the specified access register. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The ALET within the access register is used in conjunction with guest access register translation tables to locate the guest address space.

### **ASCEasce.**

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified address-space-control element (ASCE). The *asce* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional.

### **ASITasit.**

dumps the contents of absolute storage in the CP-created address space identified by the specified address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

### **ASNasn.**

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the specified address space number (ASN). The *asn* may be 1 to 4 hexadecimal digits. It is used in conjunction with guest ASN-translation tables to locate the guest address space.

### **ASTERaddr.**

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using the segment table designation contained within the specified ASN second table entry (ASTE) located at guest real address *raddr*.

## HOME

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 13.

## PRI

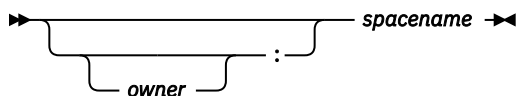
dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 1.

## SECO

dumps the contents of third-level storage (storage that appears virtual to your virtual machine) using guest control register 7.

### **SPACEspaceid.**

dumps the contents of absolute storage in the CP-created address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the CP-created address space, and *spacename* is the 1-character to 24-character address-space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

### **N**

dumps the contents of storage in hexadecimal without EBCDIC translation. This is the default. Storage is dumped in word segments. An address range is dumped starting at the fullword in which the specified address resides. Storage is dumped in fullword increments.

The storage key is dumped on the first line and at every page boundary.

### **I**

dumps the contents of storage in instruction format. The specified address is rounded down to the nearest halfword boundary.

The storage key is dumped on the first line and at every page boundary.

### **K**

dumps the storage keys in hexadecimal for all page frames or partial page frames included in the specified address range.

### **S**

dumps the contents of storage in hexadecimal translation. Storage is dumped as a string starting at the location specified. The specified address range is not boundary-aligned.

The storage key is dumped on the first line and at every page boundary.

### **T**

dumps the contents of storage in hexadecimal and character form. Storage is dumped in word segments. An address range is dumped starting at the X'10'-byte boundary in which the specified address resides. The hexadecimal words are followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

The storage key is dumped on the first line and at every page boundary.

### **U**

dumps the contents of storage in character form. Storage is dumped starting at the address specified as an EBCDIC string, or as an ASCII string if the X option is specified. The specified address range is not boundary-aligned.

The storage key is dumped on the first line and at every page boundary.

### **M**

dumps the contents of storage in hexadecimal and character form. A header is dumped as the first line, and at every page boundary and contains the high-order 32 bits of the 64-bit address along with the storage key. Storage is dumped in word segments. An address range is dumped starting at the X'10' byte boundary in which the specified address resides. The hexadecimal output is followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

### **0**

#### **hexloc1**

is the starting storage location that is to be dumped to your virtual printer. The address *hexloc1* may be 1 to 16 hexadecimal digits separated with an optional underscore for readability. Leading zeros are optional. The default is 0.

Storage locations and the minimum amount dumped are rounded according to the following conventions in hexadecimal:

Option	Start	Minimum Length
T	10	10
M	10	10
K	800	800
S	1	1
I	2	2 - 6 depending on the instruction
N	4	4
U	1	1

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the Dump command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the Dump command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**-hexloc2**

**:hexloc2**

**-END**

**:END**

is the last of a range of hexadecimal storage locations whose contents are to be dumped to your virtual printer. If specified, *hexloc2* must be equal to or greater than *hexloc1* and must be within the virtual machine storage size. The address *hexloc2* may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. A colon (:) means the same as a dash (-) when you are specifying a range of addresses.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify a dash (-) or a colon (:) without an address.

**.bytcoun**

**.END**

is a hexadecimal number designating the number of bytes of storage (starting with the byte at *hexloc1*) to be dumped to your virtual printer. If specified, the *bytcoun* must have a value of at least 1 and may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify a period (.) without a byte count value.

**BASEnn**

umps the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified general register (*nn*). The general register may be specified as a decimal number from 0 to 15 or as a hexadecimal number from 0 to F. *nn* specifies an AR/GR pair if all of the following are true:

- The L operand is specified.
- The virtual machine is an ESA virtual machine in z/Architecture mode.

## DUMP (Guest Storage - z/Architecture)

**Note:** An ESA virtual machine simulates ESA/390 architecture when initially defined. However, a guest may have the capability to switch the virtual machine from ESA/390 mode to z/Architecture mode.

- The virtual CPU is in access register mode.

The specified access register is used, in the same manner as if the *AREGareg.* operand had been specified, in conjunction with guest access-register-translation tables to locate the guest address space.

### INDEXnn

dumps the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

### \*dumpid

is an asterisk followed by a 1-character to 80-character identifier that is to be assigned to the dump. If specified, *dumpid* must be the last operand you enter on the command line.

## Usage Notes

1. The DUMP (Guest Storage) command dumps the program status words (PSWs), and the contents of registers (general, floating-point, control, and prefix), along with the contents of the specified storage. If your virtual machine is an ESA virtual machine, the subchannel information blocks (SCHIBs) and the access registers are dumped.
2. You can dump storage locations outside the range of your virtual machine size if they are within a saved segment. Any attempt to dump locations (or associated keys) between the upper limit of your virtual machine and the address at which the saved segment is saved causes the non-addressable storage response to appear in the printed output.
3. When you use the operand *ALETheyword.ALaddr.*, the primary list bit in the ALET is ignored.
4. The operands *L*, *PRI*, *SECO*, *HOME*, *ASTERaddr.*, *ASCEasce.*, *ASNasn.*, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALaddr.* use the access-register-translation (ART), the ASN translation and the dynamic-address-translation (DAT) processes. These contain special case translation processes:
  - ALET 0 always indicates the current primary address space designated by the ASCE in guest control register 1, regardless of the current contents of the dispatchable unit access list.
  - ALET 1 always indicates the current secondary address space designated by the ASCE in guest control register 7, regardless of the current contents of the dispatchable unit access list.
  - Access register 0 is always treated as containing ALET 0. The contents of access register 0 are not used during access register translation processing.
  - General register 0 is always treated as containing the value 0 when used as the base or index register. The contents of general register 0 are never used.
5. The *BASE* operand is primarily intended to be used in conjunction with the *L* operand. It lets you dump the storage used by the virtual machine in one less step than would otherwise be possible. Without the *BASE* operand, you may need to display the contents of a general register, add the displacement, and then enter the DUMP command to examine storage the program is using. With the use of access registers, the *BASE* option additionally lets you dump storage without needing to know whether your program is currently in access register mode. This is done automatically for you by CP.
6. If the *BASEnn* and *INDEXnn* operands are used together, the specified starting address for a dump is used as a displacement from the address formed from the sum of the contents of the two specified registers.

The values in the registers specified by the *BASEnn* and *INDEXnn* operands are treated as 24-bit, 31-bit or 64-bit real addresses according to the addressing mode bits (EA and BA - bits 31 and 32) in the virtual machine's PSW. The addressing mode is determined as follows:

Bit 31	Bit 32	Addressing Mode
0	0	24-bit

Bit 31	Bit 32	Addressing Mode
0	1	31-bit
1	1	64-bit

7. During DUMP command processing, the authorization mechanisms used during normal address translation are not used.
8. Special authorization is required to use the SPACE or ASIT operand of this command to dump any address space other the host-primary address space of the virtual CPU. In order to use this command to dump storage in an address space, one of the following conditions must be satisfied:
  - Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.
  - If the address space is owned by another virtual machine, your virtual machine must have read-only or read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.

Authorization is granted on a space-by-space basis, so that you may be authorized to dump one address space of a virtual machine but may not be authorized to view a different space.
9. The operands V and W used in dumping third-level storage from guest primary and secondary storage are supported for compatibility. The PRI and SECO operands replace the V and W operands.
10. When the ASIT or SPACE operand is specified, the DUMP request fails if the specified address space does not exist, or if you are not authorized to access the address space. The current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
11. The DUMP command requires an address-space-control element (ASCE) to display third-level storage (storage that appears virtual to your virtual machine). It may receive the ASCE as an operand, find it in a control register, or use the access-register translation (ART) or the address space number (ASN) translation process to obtain the ASCE. The access-register translation and ASN translation processes may fail because of one of several program exception conditions. If the translation does not succeed, the current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a response indicating the nature of the exception condition.
12. When multiple DUMP requests are entered on a single line, the BASE $nn$  and INDEX $nn$  operands are not applied to subsequent DUMP requests on the command line. They must be explicitly entered for each DUMP request.
13. If you use an underscore, it must be followed by 8 hexadecimal digits.
14. If you issue DUMP with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the DUMP command is rejected and error message HCPCDL6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CClear option
  - SET MACHine (to a different machine architecture)
15. For more information, see [“Usage Notes” on page 395](#).

## Responses

### Notes:

## DUMP (Guest Storage - z/Architecture)

1. The value *hexloc* is a hexadecimal storage address.
2. The character displayed for the variable *t* in the string *thexloc* indicates the type of storage:
  - R indicates real storage (second-level storage).
  - V indicates virtual storage (third-level storage).
3. In responses that include third-level storage, an *Rhexloc* trailer shows the second-level address that corresponds to the third-level address. The trailer is displayed on the first line of a range and at every page boundary.
4. Responses 10 and 11 appear on your display screen.

Response 1:

```
thexloc word1 word2... word8 [key] [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the T or N option. The translation is given only for the T option and is in either EBCDIC or (if the TX option is specified) ASCII. The storage key is dumped on the first line and at every page boundary.

Response 2:

```
thexlochi_ Storage Key - [key]  
hexloclo word1 word2 word3 word4 [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the M option. The translation is dumped in either EBCDIC or (if the MX option is specified) ASCII. The storage key is displayed on the first line, and at every page boundary.

Response 3:

```
thexloc *TRANSLATION* [key] [Rhexloc]
```

is the response you receive for the U option. The value *hexloc* is the location of the first data byte to be dumped. A minimum of 1 character and a maximum of 64 characters per line are dumped. The contents of storage are dumped in translated form and are in either EBCDIC or (if the UX option is specified) ASCII. Periods in your dump represent unprintable characters. The storage key is dumped on the first line of a request and at every page boundary.

Response 4:

```
thexloc mnemonic1 x-x ... mnemonic5 x-x [key] [Rhexloc]
```

is the response you receive for the I option. The value *hexloc* is the location of the first instruction to be dumped. The value *mnemonic* is a character representation of the instruction code; this is a 7-character field in which the mnemonic is left-justified. The expression *x-x* is the actual hexadecimal contents of storage and is 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. A minimum of 1 instruction and a maximum of 5 instructions are dumped per line. The storage key is displayed on the first line of a request and at every page boundary.

Response 5:

```
thexloc stringhexadecimaldata [key] [Rhexloc]
```

is the response you receive for the S option. The storage key is displayed on the first line of a request and at every page boundary.

Response 6:

```
thexloc1 TO hexloc2 KEY = key
```

is the response you receive for the K option. The values *hexloc1* and *hexloc2* are storage locations, and *key* is the associated storage key for *hexloc1*. A response is dumped for every contiguous address range that has unique storage keys, and whenever a segment boundary is crossed.

Response 7:

The PSW is used to resolve a logical address. A PSW that is not valid causes the current request to be rejected, and processing continues with the next request (if any) on the command line. In all cases, a reset PSW (all-zero) is accepted. The rejected operands are displayed along with a response indicating that the PSW is not valid. The response is:

```
addr is not addressable because of an invalid PSW.
```

where *addr* is the rejected operand or operands; it specifies an address space and a storage location within that address space.

Response 8:

The dynamic address translation (DAT) process uses the segment table designator (ASCE) to convert a third-level address to a second-level address (storage that appears real to your virtual machine). The translation process may generate one of several exceptions, identifying why one or more pages of storage are not addressable. A range of rejected storage addresses are displayed along with a response indicating the nature of the exception condition. The responses are:

```
thexloc1 to hexloc2 nonaddressable storage
- protection exception
thexloc1 to hexloc2 nonaddressable storage
- addressing exception
thexloc1 to hexloc2 nonaddressable storage
- segment translation exception
thexloc1 to hexloc2 nonaddressable storage
- page translation exception
thexloc1 to hexloc2 nonaddressable storage
- translation specification exception
thexloc1 to hexloc2 nonaddressable storage
- ASCE type exception
thexloc1 to hexloc2 nonaddressable storage
- region-first-translation exception
thexloc1 to hexloc2 nonaddressable storage
- region-second-translation exception
thexloc1 to hexloc2 nonaddressable storage
- region-third-translation exception
```

where *hexloc1* is the starting address and *hexloc2* the ending address of the non-addressable storage. The addresses take into account basing, indexing, and wrapping and are padded to 8 digits.

Response 9:

If your request includes a request to dump identical lines, you receive the following response:

```
thexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...
```

Response 10:

```
DUMPING LOC thexloc
```

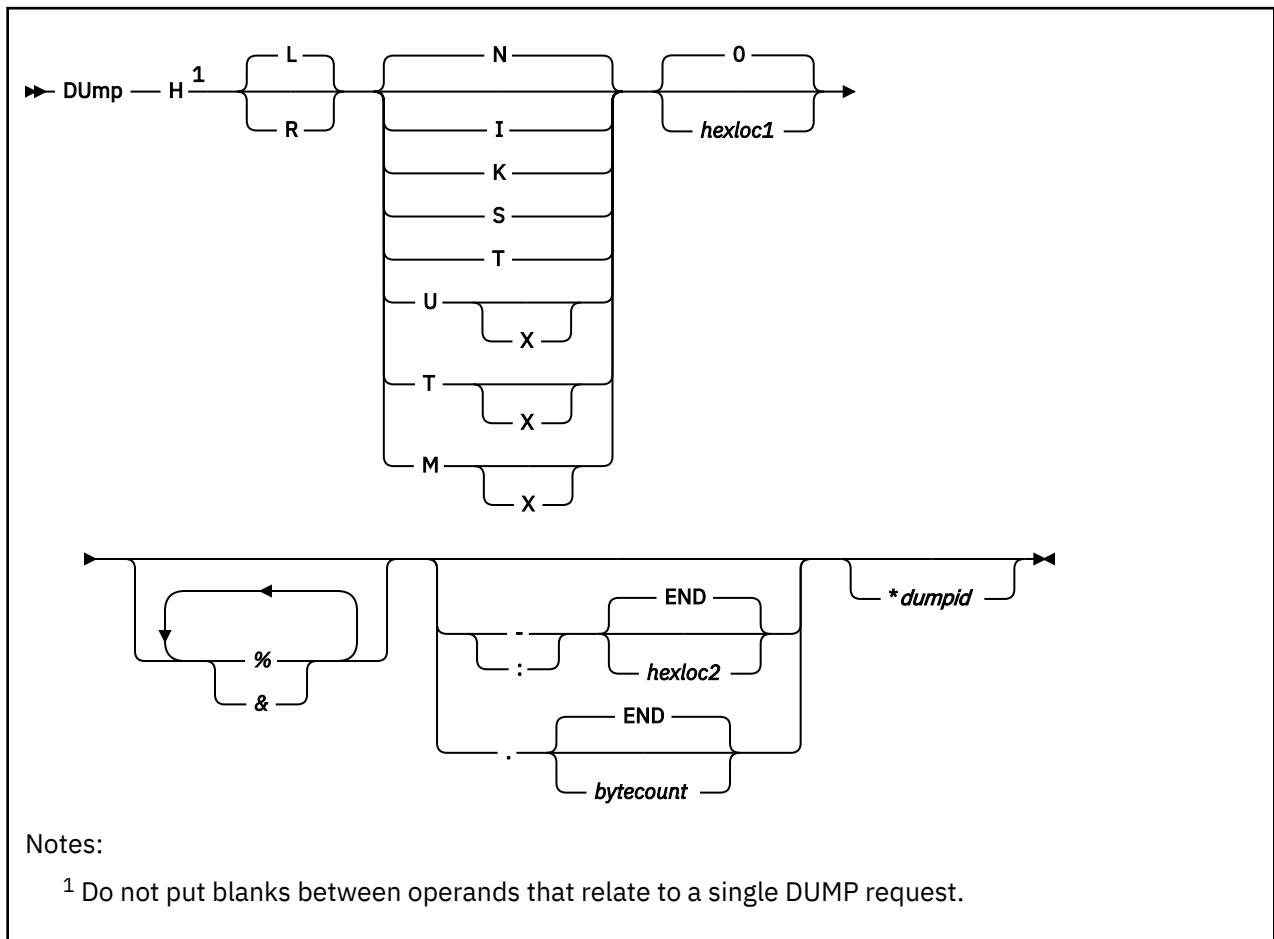
is sent to your console to indicate that a segment boundary has been reached. Each segment boundary location has an address that ends in X'00000'. The value *hexLoc* identifies the storage boundary reached.

Response 11:

```
Command complete
```

is issued when the command completes successfully.

## DUMP (Host Storage)



### Authorization

Privilege Class: C, E

### Purpose

Use DUMP (Host Storage) to dump the contents of host (first-level) storage to your virtual printer.

### Operands

**L**

specifies that host logical storage addresses are to be dumped. Host logical storage consists of 4 KB pages where each page boundary has an address that ends in X'000'. This is the default.

**R**

specifies that host real storage addresses are to be dumped. Host real storage consists of 4 KB frames where each frame boundary has an address that ends in X'000'.

**N**

.dumps the contents of storage in hexadecimal without character translation. This is the default. Storage is dumped in word segments. An address range is dumped starting at the fullword in which the specified address resides. Storage is dumped in fullword increments.

The storage key is included on the first line and at every page or frame boundary.



**I**

dumps the contents of storage in instruction format. The specified address is rounded down to the nearest halfword boundary.

The storage key is included on the first line and at every page or frame boundary.

**K**

dumps the storage keys in hexadecimal for all frames or partial frames included in the specified address range. If you are dumping host logical storage addresses, these are storage keys for the backing frames in host real storage.

**S**

dumps the contents of storage in hexadecimal translation. Storage is dumped as a string starting at the location specified. The specified address range is not boundary-aligned.

The storage key is included on the first line and at every page or frame boundary.

**T**

dumps the contents of storage in hexadecimal and character form. Storage is dumped in word segments. An address range is dumped starting at the X'10'-byte boundary in which the specified address resides. The hexadecimal words are followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

The storage key is included on the first line and at every page or frame boundary.

**U**

dumps the contents of storage in character form. Storage is dumped starting at the address specified as an EBCDIC string, or as an ASCII string if the X option is specified. The specified address range is not boundary-aligned.

The storage key is included on the first line and at every page or frame boundary.

**M**

dumps the contents of storage in hexadecimal and character form. A header is included as the first line and at every page or frame boundary. It contains the high-order 32 bits of the 64-bit address along with the storage key. Storage is dumped in word segments. An address range is dumped starting at the X'10' byte boundary in which the specified address resides. The hexadecimal output is followed by the storage as an EBCDIC string, or as an ASCII string if the X option is specified.

**0*****hexloc1***

is the starting storage location that is to be dumped. The address *hexloc1* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The default is 0.

Storage locations and the minimum amount dumped are rounded according to the following conventions in hexadecimal:

Option	Start	Minimum Length
T	10	10
M	10	10
K	800	800
S	1	1
I	2	2 - 6 depending on the instruction
N	4	4
U	1	1

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the DUMP command.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the DUMP command.

**-hexloc2**

**:hexloc2**

**-END**

**:END**

is the last of the range of hexadecimal storage locations whose contents are to be dumped. If specified, *hexloc2* must be equal to or greater than *hexloc1* and must be within the bounds of addressable storage (see usage note “3” on page 426). The address *hexloc2* may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. A colon (:) means the same as a dash (-) when you are specifying a range of addresses.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify a dash (-) or a colon (:) without an address.

**.bytecount**

**.END**

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at *hexloc1*) to be dumped. If specified, the *bytecount* must have a value of at least 1 and may be from 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The sum of *hexloc1* and *bytecount* must be an address within the bounds of addressable storage (see usage note “3” on page 426).

Specify END if you want to dump the remaining contents of storage. END is the default if you specify a period (.) without a byte count value.

**\*dumpid**

is an asterisk followed by a 1-character to 97-character identifier (truncated to 80 characters for a z/Architecture guest), that is to be assigned to the dump. If specified, *dumpid* must be the last operand you enter on the command line.

## Usage Notes

1. While dumping host storage, the running of a virtual machine or someone using the STORE (Host Storage) command could cause data to be changed. This could create an inconsistency in the data that is viewed.
2. In a multiprocessor environment, each processor has a prefix register that it uses to relocate addresses between 0 and X'1FFF' (an 8 KB area known as the prefix area) to two other contiguous frames in real storage. The prefix register enables each processor to use a different prefix area to avoid conflict with other processors for such activity as interrupt code recording. Thus, the range 0 through X'1FFF' refers to different areas of storage, depending on which processor generates the address. All references to first-level storage are handled as if they were made on the IPL processor. Because those locations in the system execution space are identity-mapped, addresses in the range 0 through X'1FFF' have the same content in DUMP HL output and DUMP HR output.
3. For host logical storage (the system execution space), addressable storage includes the system execution area, which resides below 2 GB, plus the system frame table and the system execution space page management table, which reside above 2 GB. To display the storage address ranges used for the system execution space, issue the QUERY SXSSTORAGE command. The address range that you specify on the DUMP command must start and end in the same address range of the system execution space. If the END operand is specified (or allowed to default), it refers to the end of the range in which the starting address is located.

For host real storage, addressable storage is the lesser of CP's configured storage size (if specified at system generation) and the real machine storage size. To display the size of host real storage, issue the QUERY STORAGE command.

4. When dumping host logical storage, if you specify an address that is not backed with a real storage frame, error message HCP6151E is issued, even if the address is included in the ranges of in-use and usable addresses displayed by the QUERY SXSSORAGE command.
5. If you use an underscore, it must be followed by 8 hexadecimal digits.
6. For more information, see [“Usage Notes” on page 395](#).

## Responses

### Notes:

1. The variable *hexloc* represents a hexadecimal storage address. The variables *hexlochi* and *hexloclo* represent the high-order 32 bits and low-order 32 bits of a 64-bit hexadecimal storage address.
2. The character displayed for the variable *t* in the string *Hthexloc* indicates the type of storage:
  - L indicates host logical storage.
  - R indicates host real storage.
3. In responses that include host logical storage, an *Rhexloc* trailer shows the host real storage address that corresponds to the host logical storage address. The trailer is displayed on the first line of a range and at every page boundary.
4. Responses 8 and 9 appear on your display screen.

### Response 1:

```
Hthexloc word1 word2 word3 word4 [key] [*TRANSLATION*] [Rhexloc]
```

is the response you receive for the T or N option. The translation is dumped only if you have also entered the T option and is in either EBCDIC or (if the TX option is specified) ASCII. The storage key is dumped on the first line, and at every page or frame boundary.

### Response 2:

```
Hthexlochi_ Storage Key - [key]  
hexloclo word1 word2 word3 word4 [*TRANSLATION*] [Lhexloc] [Rhexloc]
```

is the response you receive for the M option. The translation is displayed in either EBCDIC or (if the MX option is specified) ASCII. The storage key is displayed on the first line, and at every page or frame boundary.

### Response 3:

```
Hthexloc *TRANSLATION* [key] [Rhexloc]
```

is the response you receive for the U option. The value *hexloc* is the location of the first data byte to be dumped. A minimum of 1 character and a maximum of 64 characters are dumped per line. The contents of storage are dumped in translated form and are in either EBCDIC or (if the UX option is specified) ASCII. Periods represent unprintable characters. The storage key is dumped on the first line and at every page or frame boundary.

### Response 4:

```
Hthexloc mnemonic1 x-x ... mnemonic5 x-x [key] [Rhexloc]
```

is the response you receive for the I option. The value *hexloc* is the location of the first instruction to be dumped. The value *mnemonic* is a character representation of the instruction code; this is a 7-character field in which the mnemonic is left-justified. The expression *x-x* is the actual hexadecimal contents of storage and is 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction

## DUMP (Host Storage)

codes are indicated by a series of asterisks. A minimum of 1 instruction and a maximum of 5 instructions are dumped per line. The storage key is dumped on the first line and at every page or frame boundary.

Response 5:

```
Hthexloc stringhexadecimaldata [key] [Rhexloc]
```

is the response you receive for the S option. The storage key is dumped on the first line and at every page or frame boundary.

Response 6:

```
Hthexloc1 to hexloc2 KEY = key
```

is the response you receive for the K option. The values *hexloc1* and *hexloc2* are storage locations and *key* is the associated storage key for *hexloc1*. You receive this response for every contiguous address range that has unique storage keys, and whenever a segment boundary is crossed.

Response 7:

If your request includes a request to dump one or more lines identical with the previous line, you receive the following response:

```
Hthexloc1 to hexloc2 suppressed line(s) same as above ...
```

Response 8:

```
dumping LOC Hthexloc
```

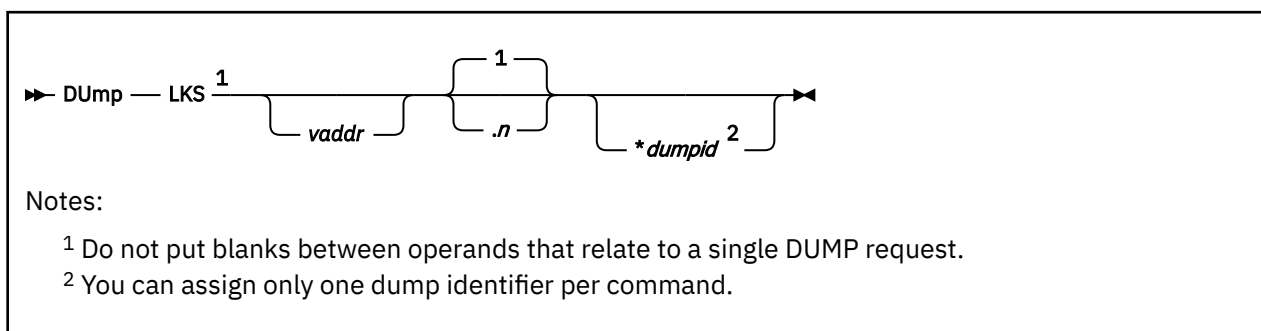
is sent to your console display screen to indicate that a segment boundary has been reached. Each segment boundary has an address that ends in X'00000'.

Response 9:

```
Command complete
```

is sent to your console display screen when the command completes successfully.

## DUMP LKS



### Authorization

Privilege Class: G

### Purpose

Use DUMP LKS to dump the state entries of the guest linkage stack.

### Operands

#### *vaddr*

dumps the branch or program-call linkage-stack state entry that contains the specified guest virtual address within the current home address space designated by guest control register 13. The default is the current linkage-stack state entry pointed to by guest control register 15.

#### **1**

#### *n*

is a decimal number, up to eight digits, specifying the number of branch and program-call linkage-stack state entries to be dumped. The dump starts with the specified/current entry and moves back toward the first entry on the linkage-stack. The default is 1, and 0 is treated as 1.

#### **\*dumpid**

is an asterisk followed by a 1-character to 97-character identifier that is to be assigned to the dump (may be truncated to 80 characters for a z/Architecture guest). If specified, *\*dumpid* must be the last operand you enter on the command line.

### Usage Notes

1. This command is valid only if the virtual machine is an ESA virtual machine.
2. The *first* linkage-stack state entry is the oldest state entry (the first one put on the stack). The *last* linkage-stack state entry is the current state entry (the most recent one put on the stack).
3. Unlike other DUMP command operands, using LKS does not define the default for subsequent operands of the current DUMP function.
4. The virtual address specified in *vaddr* must fall within the branch or program-call linkage-stack state entry, including the first and last bytes of the linkage-stack state entry.
5. A linkage-stack state entry is an entry on the linkage stack that describes the environment in an address space (such as access registers, general-purpose registers and PSWs).
6. For more information, see [“Usage Notes” on page 395](#).

## Responses

Heading	Description
ED-ADDR	The linkage-stack-entry description address, a guest home virtual address represented as eight hexadecimal digits (16 hexadecimal digits for a z/Architecture guest).
U	The unstack-suppression bit, either 0 or 1
ET	The entry type, either BRN for a branch state entry or PRC for a program-call state entry.
SI	The section identification, represented as two hexadecimal digits.
RFS	The remaining free space, represented as four hexadecimal digits.
NES	The next-entry size, represented as four hexadecimal digits.
PKM	The PSW-key mask, represented as four hexadecimal digits.
SASN	The secondary ASN, represented as four hexadecimal digits.
EAX	The extended authorization index, represented as four hexadecimal digits.
PASN	The primary ASN, represented as four hexadecimal digits.
PSW	The program-status word, represented as 16 hexadecimal digits (32 hexadecimal digits for a z/Architecture guest).
A	The addressing mode, which may be either a 0 or 1 for an ESA/390 mode guest or 24, 31, or 64 for a z/Architecture guest.
B-ADDR	The branch address, represented as eight hexadecimal digits.
MOD-AREA	The modifiable area, represented as 16 hexadecimal digits.
PC-NUM	The program call number, represented as eight hexadecimal digits.
CSID	The called-space identification, if any, represented as eight hexadecimal digits.
SINST	Secondary ASTE instance number, if any, represented as eight hexadecimal digits.
PIINST	Primary ASTE instance number, if any, represented as eight hexadecimal digits.
GPR	General purpose register, represented as eight hexadecimal digits.
GRG	General purpose register for z/Architecture guests, represented as 16 hexadecimal digits.
AR	Access register, represented as eight hexadecimal digits.

### Response 1:

The linkage-stack entry for an ESA/390 branch state entry has the following format:

```

ED-ADDR  U  ET  SI  RFS  NES  PKM  SASN  EAX  PASN
nnnnnnnn n  BRN nn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn

          PSW          A  B-ADDR          MOD-AREA
nnnnnnnn nnnnnnnn  n  nnnnnnnn  nnnnnnnnnnnnnnnn

GPR  0 = greg0  greg1  greg2  greg3
GPR  4 = greg4  greg5  greg6  greg7
GPR  8 = greg8  greg9  greg10 greg11
GPR 12 = greg12 greg13 greg14 greg15

AR   0 = areg0  areg1  areg2  areg3
AR   4 = areg4  areg5  areg6  areg7
AR   8 = areg8  areg9  areg10 areg11
AR  12 = areg12 areg13 areg14 areg15

```

The headings in this response are described under “Responses”.

Response 2:

The linkage-stack entry for a z/Architecture branch state entry has the following format:

```

ED-ADDR      U  ET  SI  RFS  NES  PKM  SASN  EAX  PASN
nnnnnnnnnn  n  BRN  nn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn

                PSW                      A          B-ADDR
nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn  nn  nnnnnnnnnnnnnnnnn

                MOD-AREA      SINST      PINST
nnnnnnnnnnnnnnnnnn  nnnnnnnn  nnnnnnnn

GRG  0 = greg0          greg1
GRG  2 = greg2          greg3
GRG  4 = greg4          greg5
GRG  6 = greg6          greg7
GRG  8 = greg8          greg9
GRG 10 = greg10         greg11
GRG 12 = greg12         greg13
GRG 14 = greg14         greg15

AR   0 = areg0  areg1  areg2  areg3
AR   4 = areg4  areg5  areg6  areg7
AR   8 = areg8  areg9  areg10 areg11
AR  12 = areg12 areg13 areg14 areg15

```

The headings in this response are described under “Responses”.

ED-ADDR and B-ADDR can be either 32 or 64 bits.

Response 3:

The linkage-stack entry for an ESA/390 program call state entry has the following format:

```

ED-ADDR  U  ET  SI  RFS  NES  PKM  SASN  EAX  PASN
nnnnnnnn  n  PRC  nn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn

                PSW      PC-NUM      MOD-AREA      CSID
nnnnnnnnn  nnnnnnnn  nnnnnnnnnnnnnnnnn  nnnnnnnnn

GPR  0 = greg0  greg1  greg2  greg3
GPR  4 = greg4  greg5  greg6  greg7
GPR  8 = greg8  greg9  greg10  greg11
GPR 12 = greg12  greg13  greg14  greg15

AR   0 = areg0  areg1  areg2  areg3
AR   4 = areg4  areg5  areg6  areg7
AR   8 = areg8  areg9  areg10  areg11
AR  12 = areg12  areg13  areg14  areg15

```

The headings in this response are described under “Responses”.

Response 4:

The linkage-stack entry for a z/Architecture program call state entry has the following format:

```

ED-ADDR      U  ET  SI  RFS  NES  PKM  SASN  EAX  PASN
nnnnnnnnnn  n  PRC  nn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn

                PSW                      PC-NUM
nnnnnnnnn  nnnnnnnn  nnnnnnnnnnnnnnnnn

                MOD-AREA      CSID      SINST      PINST
nnnnnnnnnnnnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn

GRG  0 = greg0          greg1
GRG  2 = greg2          greg3
GRG  4 = greg4          greg5
GRG  6 = greg6          greg7
GRG  8 = greg8          greg9
GRG 10 = greg10         greg11
GRG 12 = greg12         greg13
GRG 14 = greg14         greg15

AR   0 = areg0  areg1  areg2  areg3
AR   4 = areg4  areg5  areg6  areg7

```

```
AR 8 = areg8 areg9 areg10 areg11
AR 12 = areg12 areg13 areg14 areg15
```

The headings in this response are described under “Responses”.

ED-ADDR can be either 32 or 64 bits.

Response 5:

If a guest virtual address from the home address space cannot be converted to a host address, the translation process generates one of several exceptions, identifying the first page that was not addressable. The responses are:

```
Vhexloc1 to hexloc2 nonaddressable storage
- protection exception
Vhexloc1 to hexloc2 nonaddressable storage
- addressing exception
Vhexloc1 to hexloc2 nonaddressable storage
- segment translation exception
Vhexloc1 to hexloc2 nonaddressable storage
- page translation exception
Vhexloc1 to hexloc2 nonaddressable storage
- translation specification exception
```

The operand *Vhexloc1* is the starting address and *hexloc2* is the ending address of the nonaddressable storage.

Response 6:

If a state entry that is neither a branch nor a program-call is reached, the following response is issued:

```
Vhexloc1 to hexloc2 are linkage stack state entries that are
not branch or program-call.
```

The operand *Vhexloc1* is the starting address and *hexloc2* the ending address of the block of contiguous state entries in a linkage stack section that are not branch or program calls.

Response 7:

```
The linkage stack is empty.
```

is issued if the linkage stack is empty.

Response 8:

```
Virtual address hexloc is not within a linkage
stack branch state entry or a program-call state entry.
```

is issued if the specified guest virtual address is not within a branch or program-call state entry. The operand *Hexloc* is the guest virtual address specified on the command line. Header and trailer state entries are not displayed.

Response 9:

```
The first state entry in the linkage stack has been reached.
```

is issued if the bottom of the stack is reached during command processing.



## DUMP PREFIX



### Authorization

Privilege Class: G

### Purpose

Use DUMP PREFIX to send prefix register contents to your virtual printer.

### Operands

#### *\*dumpid*

identifies a 1- to 97-character string that is to be assigned to the dump. If specified, *\*dumpid* must be the last operand you enter on the command line.

### Usage Notes

1. The prefix register contains the absolute location of guest page 0.
2. For more information, see [“Usage Notes” on page 395](#).

### Responses

Response 1:

```
PREFIX = xxxxxxxx
```

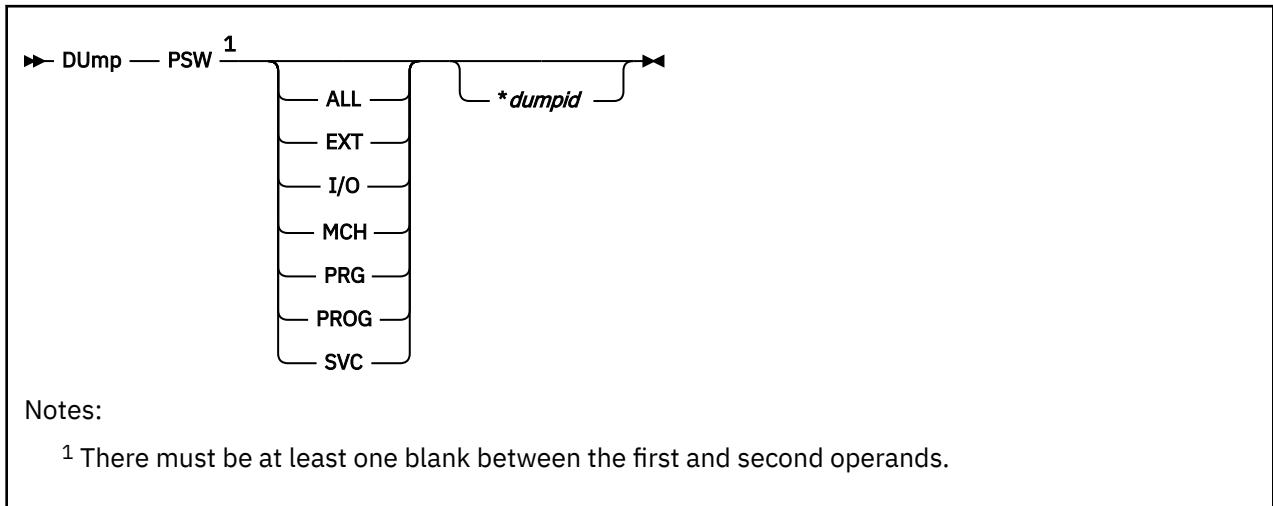
is the hexadecimal content of the prefix register.

Response 2:

```
Command complete
```

is issued when the command completes successfully.

## DUMP PSW



### Authorization

Privilege Class: G

### Purpose

Use DUMP PSW to send the program status word to your virtual printer.

### Operands

#### ALL

dumps the current values and appropriate interrupt information for the current PSW and all old and new interrupt PSWs (except restart).

#### EXT

#### I/O

#### MCH

#### PRG

#### PROG

#### SVC

dumps the current values and appropriate interrupt information for both the new and old PSWs specified—that is, SVC, program check, machine check, I/O, and external.

#### *\*dumpid*

identifies a 1-character to 97-character string that is to be assigned to the dump (may be truncated to 80 characters for a z/Architecture guest). If specified, *\*dumpid* must be the last operand you enter on the command line.

### Usage Notes

1. The DUMP PSW command is valid for a z/Architecture virtual machine only if SET PSWTRANS (with ALL or DISPLAY option) has been issued previously.
2. DUMP PSW ALL dumps the PSW as a 128-bit PSW for z/Architecture and z/XC virtual machines.
3. z/Architecture and z/XC expands the PSW to 128 bits to accommodate 64-bit instruction addresses. The expanded PSWs are referenced in the command output as "PSWG".

4. The SET PSWTRANS command formats the 128-bit PSWs from a z/Architecture or z/XC virtual machine to the 64-bit format that can be displayed and manipulated in an ESA/390 or ESA/XC virtual machine.

When the SET PSWTRANS and DUMP PSW commands are issued from a z/Architecture or z/XC virtual machine, the 128-bit format PSW is converted to a 64-bit format PSW by the following changes:

- Bit 12 of the 64-bit PSW is set to 1.
- Bit 31 of the 64-bit PSW is set to 0.
- Bits 0-11 and 13-30 of the 64-bit PSW are set to the value of the corresponding bits of the 128-bit PSW.
- Bits 33-63 of the 64-bit PSW are set to bits 97-127 of the 128-bit PSW.
- If bits 64-95 of the 128-bit PSW are not 0, then the PSW is flagged as non-translatable.
- Bit 32 is set as follows:
  - If bits 31 and 32 of the 128-bit PSW are both 1, then the PSW is flagged as non-translatable.
  - If bits 31 and 32 of the 128-bit PSW are both 0, then bit 32 of the 64-bit PSW is set to 0.
  - If bit 31 of the 128-bit PSW is 0 and bit 32 is 1, then bit 32 of the 64-bit PSW is set to 1.

## Responses

The response to all DUMP options is "Command complete". What follows is a description of the dump information in the generated print file.

Response 1:

The command DUMP PSW with no other options formats the current PSW for a print file. This is the response you receive on ESA/XC and ESA/390 virtual machines:

```
PSW = 03E40000 8003010C
```

Response 2:

The command DUMP PSW with no other options formats the current PSW for a print file. This is the response that you receive on z/Architecture and z/XC virtual machines:

```
PSW = 03EC0000 8003010C  PSWG = 03E40000 80000000 00000000 0003010C
```

Response 3:

The command DUMP PSW with no other options formats the current PSW for a print file. If the current PSW for a guest machine cannot be converted to a 64-bit PSW, then this is the response you receive on z/Architecture virtual machines:

```
PSW = NON TRANSLATABLE  PSWG = 03E40000 80000000 00000000 0003010C
```

Response 4:

The command DUMP PSW ALL formats the old and new PSWs and expands the interruption code into four hexadecimal digits in front of the old PSWs. *nnnn* represents the EC-mode interruption code. The numbers 18, 20, 28... are storage locations. Each *xxxxxxx xxxxxxxx* represents the content of a PSW. This is the response you receive on ESA/XC and ESA/390 virtual machines:

```
PSW = xxxxxxxx xxxxxxxx
EXT nnnn 18 OLD xxxxxxxx xxxxxxxx 58 NEW xxxxxxxx xxxxxxxx
SVC nnnn 20 OLD xxxxxxxx xxxxxxxx 60 NEW xxxxxxxx xxxxxxxx
PRG nnnn 28 OLD xxxxxxxx xxxxxxxx 68 NEW xxxxxxxx xxxxxxxx
MCH nnnn 30 OLD xxxxxxxx xxxxxxxx 70 NEW xxxxxxxx xxxxxxxx
I/O nnnn 38 OLD xxxxxxxx xxxxxxxx 78 NEW xxxxxxxx xxxxxxxx
```

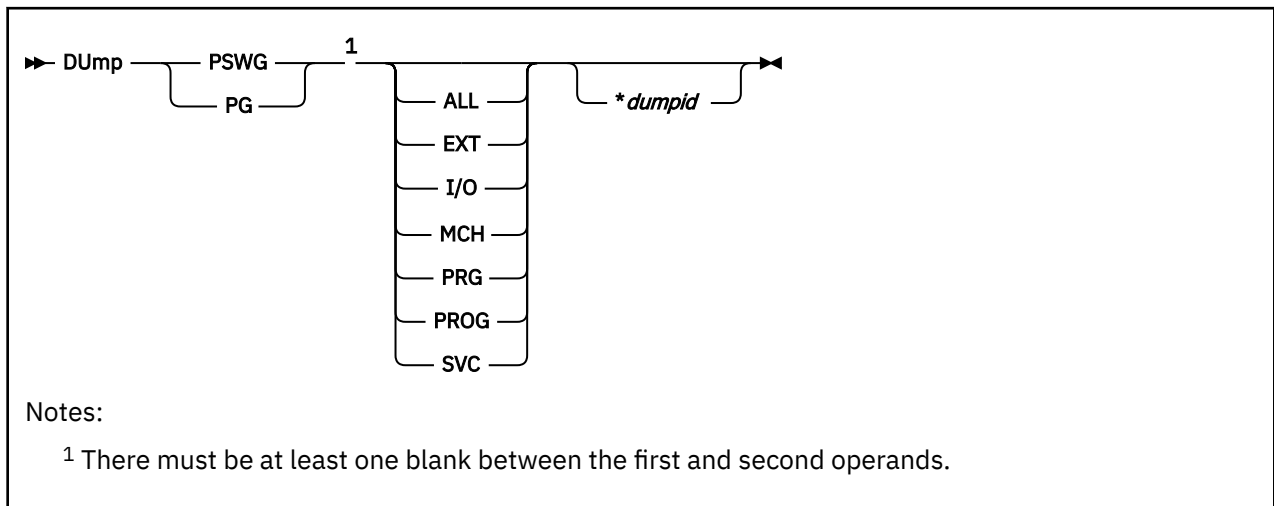
Response 5:

## DUMP PSW

The command DUMP PSW ALL formats the old and new PSWs and expands the interruption code into four hexadecimal digits in front of the old PSWs. *nnnn* represents the interruption code. The numbers 130, 140, 150, ... are storage locations. Each *xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx* represents the content of a PSW. This is the response that you receive on z/Architecture and z/XC virtual machines:

```
PSW = 03EC0000 8003010C   PSWG = 03E40000 80000000 00000000 0003010C
EXT nnnn 130 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1B0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
SVC nnnn 140 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1C0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
PRG nnnn 150 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1D0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
MCH nnnn 160 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1E0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
I/O nnnn 170 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1F0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

## DUMP PSWG



### Authorization

Privilege Class: G

### Purpose

Use DUMP PSWG to send the program status word of your z/Architecture or z/XC virtual machine to your virtual printer.

### Operands

#### ALL

dumps the current values and appropriate interrupt information for the current PSW and all old and new interrupt PSWs (except restart).

#### EXT

#### I/O

#### MCH

#### PRG

#### PROG

#### SVC

dumps the current values and appropriate interruption information for both the new and old PSWs specified—that is, SVC, program check, machine check, I/O, or external.

#### *\*dumpid*

identifies a 1- to 80-character string that is to be assigned to the dump. If specified, *\*dumpid* must be the last operand you enter on the command line.

### Usage Notes

1. To dump the PSW for an ESA/390 or ESA/XC virtual machine, use the DUMP PSW command.

### Responses

The response to all DUMP options is "Command complete". What follows is a description of the dump information in the generated print file.

Response 1:

## DUMP PSWG

The command DUMP PSWG with no other options formats the current PSW for a print file.

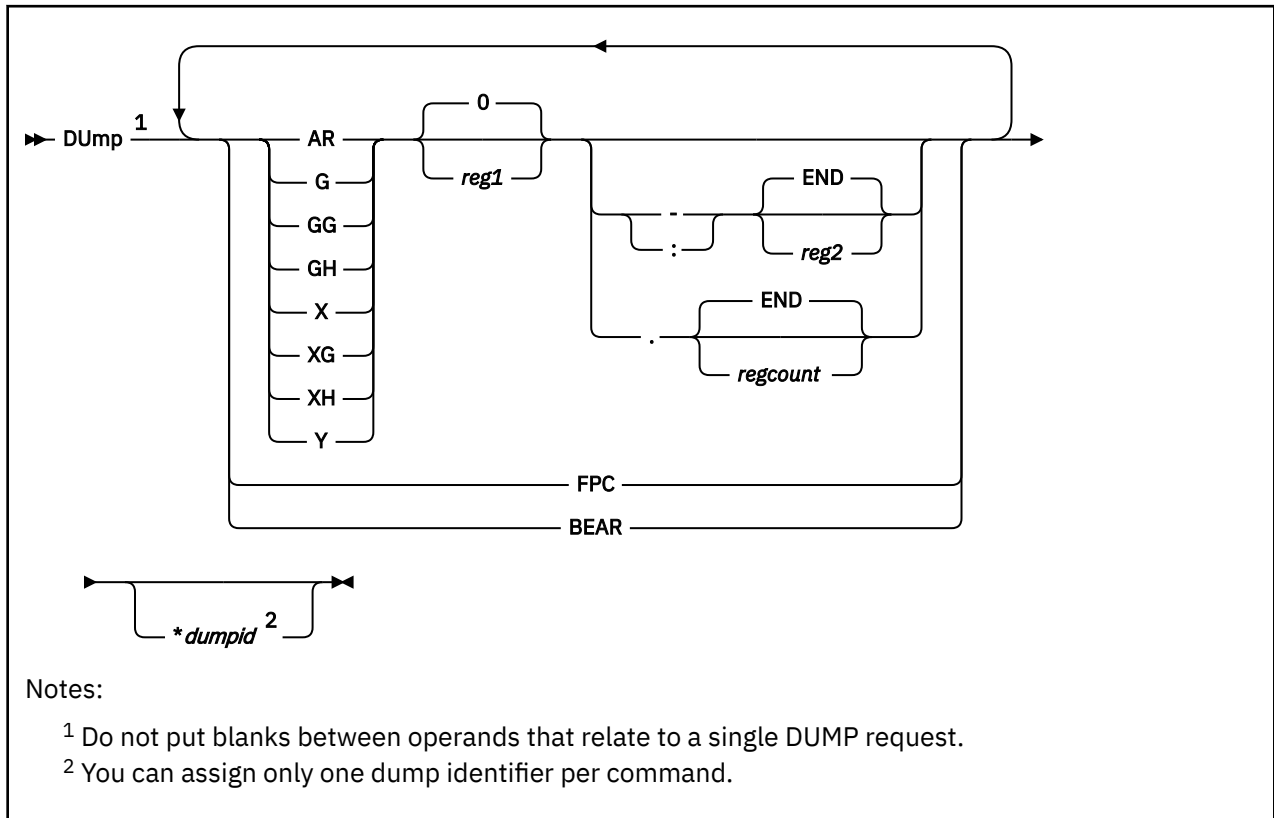
```
PSW = 03E40000 80000000 00000000 0003010C
```

Response 2:

The command DUMP PSWG ALL formats the old and new PSWs and expands the interruption code into four hexadecimal digits in front of the old PSWs. *nnnn* represents the interruption code. The numbers 130, 140, 150, ... are storage locations. Each *xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx* represents the content of a PSW.

```
PSW = 03E40000 80000000 00000000 0003010C
EXT nnnn 130 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1B0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
SVC nnnn 140 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1C0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
PRG nnnn 150 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1D0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
MCH nnnn 160 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1E0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
I/O nnnn 170 OLD xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
      1F0 NEW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

## DUMP (Registers)



### Authorization

Privilege Class: G

### Purpose

Use DUMP (Registers) to dump the following types of registers to your virtual printer:

- Access
- Control
- Floating-point
- Floating-point control
- General-purpose

### Operands

**0**

#### **AR***reg1*

is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, access register whose contents are to be dumped to your virtual printer. If you specify AR without a register number, the contents of all the access registers are dumped.

#### **G***reg1*

G indicates a full register for an ESA/390 or ESA/XC guest or the low half of the register for a z/Architecture or z/XC guest. *reg1* is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be dumped to your virtual

## DUMP (Registers)

printer. If you specify G without a register number, the appropriate contents of all the general registers are dumped.

### **GGreg1**

GG indicates a full register (32 bits for an ESA/390 or ESA/XC guest, 64 bits for a z/Architecture or z/XC guest). *reg1* is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be dumped to your virtual printer. If you specify GG without a register number, the appropriate contents of all the general registers are dumped.

### **GHreg1**

GH indicates the upper half of a register for a z/Architecture or z/XC guest. This option is not valid for an ESA/390 or ESA/XC guest. *reg1* is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be dumped to your virtual printer. If you specify GH without a register number, the high halves of all of the general registers are dumped.

### **Xreg1**

X indicates a full control register for an ESA/390 or ESA/XC guest or the low half of the control register for a z/Architecture or z/XC guest. *reg1* is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, control register whose contents are to be dumped to your virtual printer. If you specify X without a register number, the appropriate contents of all the control registers are dumped.

### **XGreg1**

XG indicates a full register (32 bits for an ESA/390 or ESA/XC guest, 64 bits for a z/Architecture or z/XC guest). *reg1* is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, control register whose contents are to be dumped to your virtual printer. If you specify XG without a register number, the appropriate contents of all the control registers are dumped.

### **XHreg1**

XH indicates the upper half of a register for a z/Architecture or z/XC guest. This option is not valid for an ESA/390 or ESA/XC guest. *reg1* is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, control register whose contents are to be dumped to your virtual printer. If you specify XH without a register number, the high halves of all of the control registers are dumped.

### **Yreg1**

*reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, floating-point register whose contents are to be dumped. If you specify Y without a register number, the contents of all the floating-point registers are dumped.

### **FPC**

FPC dumps the contents of the floating-point register.

### **BEAR**

BEAR dumps the contents of the breaking-event-address register if the PER-3 facility is available.

### **: END**

#### **- reg2**

*reg2* is a number of the last register whose contents are to be dumped. You must specify the dash (-) or colon (:) when you are requesting more than one register by register number. You must specify the range of registers in ascending order. Therefore, the value you specify for *reg2* must be equal to or greater than the value you specify for *reg1*.

The colon (:) means the same as a dash (-) when you are specifying a range of addresses.

If you specify AR, G, GG, GH, X, XG, XH, or Y, *reg2* may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F. The contents of register 1 through register 2 are dumped to your virtual printer.

Specify END to dump the contents of all remaining registers. END is the default.



**END****. regcount**

*regcount* is a decimal number from 1 to 16 or a hexadecimal number from 1 to F specifying the number of registers (starting with register 1) whose contents are to be dumped. If you specify AR, G, GG, GH, X, XG, XH, or Y, *regcount* can be a decimal number from 1 to 16 or a hexadecimal number from 1 to F.

The sum of *reg1* and *regcount* must be a number that does not exceed the maximum register number for the type of registers being dumped.

Specify END to dump the contents of all remaining registers. END is the default.

**\*dumpid**

is an asterisk followed by a 1- to 97- character identifier that is to be assigned to the dump (may be truncated to 80 characters for a z/Architecture guest). If specified, *\*dumpid* must be the last operand you enter on the command line.

**Usage Notes**

1. The G and X operands are valid for an ESA/390, ESA/XC, z/Architecture, or z/XC guest. For an ESA/390 or ESA/XC guest, these operands refer to the full four-byte register. For a z/Architecture or z/XC guest, these operands refer to the low-order four bytes of the full eight-byte register.
2. The GG and XG operands are valid for an ESA/390, ESA/XC, z/Architecture, or z/XC guest. For an ESA/390 or ESA/XC guest, these operands refer to the full four-byte register. For a z/Architecture or z/XC guest, these operands refer to the full eight-byte register.
3. The GH and XH operands are valid only for a z/Architecture or z/XC guest. These operands refer to the high-order four bytes of the full eight-byte register.
4. When a dash (-), colon (:), or period (.) is not preceded by a register number, the first register number defaults to 0. When a dash, colon, or period is not followed by a register number, the last register defaults to END.
5. For more information, see [“Usage Notes” on page 395](#).

**Responses**

Response 1:

Dump Register

```
GPR nn = genreg0 genreg1 genreg2 genreg3
```

is the response that is sent to the virtual printer when you dump the general registers using *D G* or *D GG* from an ESA/390 or ESA/XC guest. *nn* indicates the first (or only) register whose contents are dumped.

The content of each register is dumped in hexadecimal. Up to four registers per line are dumped for each range you specify. If required, multiple lines are dumped to a maximum of four lines.

Response 2:

```
HCPCDE6150E An option has been specified that is not valid in
ESA/390 mode.
```

is the response you receive when you dump general registers using *D GH* from an ESA/390 guest (an appropriate message will be issued from an ESA/XC guest).

Response 3:

Dump Register

```
GPR nn = genreg0l genreg1l genreg2l genreg3l
```

is the response that is sent to the virtual printer when you dump the general registers using *D G* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are dumped.

## DUMP (Registers)

The content of each register is dumped in hexadecimal and represents the low half of the full 64-bit register. Up to four registers per line are dumped for each range you specify. If required, multiple lines are dumped to a maximum of four lines.

Response 4:

Dump Register

```
GRG nn = genreg0hgenreg0l genreg1hgenreg1l
```

is the response that is sent to the virtual printer when you dump the general registers using *D GG* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are dumped.

The content of each register is dumped in hexadecimal and represents the full 64-bit register. Up to two registers per line are dumped for each range you specify. If required, multiple lines are dumped to a maximum of eight lines.

The high and low halves of each register may be separated by an optional underscore for readability. To have the underscore displayed, use the SET UNDERSCORE command.

Response 5:

Dump Register

```
GRH nn = genreg0h genreg1h genreg2h genreg3h
```

is the response that is sent to the virtual printer when you dump the general registers using *D GH* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are dumped.

The content of each register is dumped in hexadecimal and represents the high half of the full 64-bit register. Up to four registers per line are dumped for each range you specify. If required, multiple lines are dumped to a maximum of four lines.

Response 6:

Dump Register

```
ECR nn = ctlreg0 ctlreg1 ctlreg2 ctlreg3
```

is the response that is sent to the virtual printer when you dump the control registers using *D X* or *D XG* from an ESA/390 or ESA/XC guest. *nn* indicates the first (or only) register whose contents are dumped.

The content of each register is dumped in hexadecimal. Up to four registers per line are dumped for each range you specify. If required, multiple lines are dumped to a maximum of four lines.

Response 7:

```
HCPCDE6150E An option has been specified that is not valid in  
ESA/390 mode.
```

is the response you receive when you dump the control registers using *D XH* from an ESA/390 guest (an appropriate message will be issued from an ESA/XC guest).

Response 8:

Dump Register

```
ECR nn = ctlreg0l ctlreg1l ctlreg2l ctlreg3l
```

is the response that is sent to the virtual printer when you dump the control registers using *D X* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are dumped.

The content of each register is dumped in hexadecimal and represents the low half of the full 64-bit register. Up to four registers per line are dumped for each range you specify. If required, multiple lines are dumped to a maximum of four lines.

Response 9:

## Dump Register

```
CRG nn = ctlreg0hctlreg0l ctlreg1hctlreg1l
```

is the response that is sent to the virtual printer when you dump the control registers using *D XG* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are dumped.

The content of each register is dumped in hexadecimal and represents the the full 64-bit register. Up to two registers per line are dumped for each range you specify. If required, multiple lines are dumped to a maximum of eight lines.

The high and low halves of each register may be separated by an optional underscore for readability. To have the underscore displayed, use the SET UNDERSCORE command.

Response 10:

## Dump Register

```
CRH nn = ctlreg0h ctlreg1h ctlreg2h ctlreg3h
```

is the response that is sent to the virtual printer when you dump the control registers using *D XH* from a z/Architecture or z/XC guest. *nn* indicates the first (or only) register whose contents are dumped.

The content of each register is dumped in hexadecimal and represents the high half of the full 64-bit register. Up to four registers per line are dumped for each range you specify. If required, multiple lines are dumped to a maximum of four lines.

Response 11:

## Dump Register

```
AR nn = areg0 areg1 areg2 areg3
```

is the response that is sent to the virtual printer when you dump the access registers; *nn* is the first (or only) access register whose contents are dumped.

The contents of the requested access registers are dumped in hexadecimal. Up to four registers per line are dumped. If required, multiple response lines are dumped to a maximum of four lines.

Response 12:

## Dump Register

```
FPR nn = xxxxxxxxxxxxxxxxxxx ±.ddddddddddddddd E dd
```

is the response that is sent to the virtual printer when you dump the floating-point registers. The *nn* is the floating-point register whose contents are dumped on the line. The contents of the floating-point registers are dumped in both the internal hexadecimal format and the E-format; the first part of the response is the contents of the register in hexadecimal, and the second part of the response is the contents in exponential decimal format. Only one register is dumped per line. If you specified a range of registers, multiple response lines are dumped.

Response 13:

## Dump Register

```
FPC = mmmmmmm
```

is the response that is sent to the virtual printer when you dump the floating-point control register. The contents of the floating-point control register are dumped in hexadecimal format.

Response 14:

## Dump Register

```
BEA = mmmmmmmmmmmmmmm
```

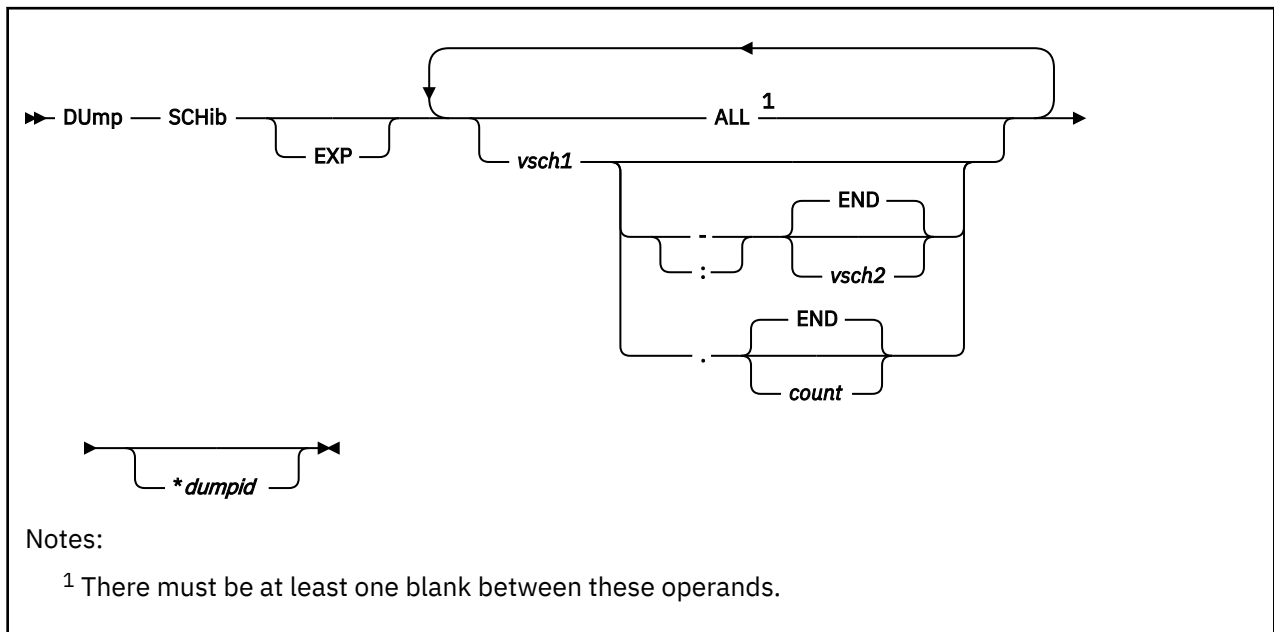
## **DUMP (Registers)**

is the response that is sent to the virtual printer when you dump the breaking-event-address register. The contents of the breaking-event-address register are in hexadecimal format.

For ESA/390 and ESA/XC mode guests, the breaking event address contains 4 bytes. In z/Architecture mode, the breaking event address contains 8 bytes.

In z/Architecture mode the high and low half of the register might be separated by an optional underscore for readability. To display the underscore, use the SET UNDERSCORE command.

## DUMP SCHIB



### Authorization

Privilege Class: G

### Purpose

Use DUMP SCHIB to send information from your subchannel information blocks (SCHIBs) to your virtual printer.

### Operands

#### EXP

specifies that extended channel measurement information is to be displayed.

#### ALL

specifies that SCHIBs for all virtual devices are to be dumped.

#### *vsch1*

*vsch2* identifies the starting (or only) virtual subchannel whose SCHIB is to be dumped.

#### : END

#### - *vsch2*

*vsch2* specifies that all SCHIBs in the range *vsch1* to *vsch2* are to be dumped. Specify END to dump all SCHIBs from *vsch1* to the highest number subchannel. END is the default.

A colon (:) means the same as a hyphen (-) when specifying a range.

#### END

#### . *count*

*count* specifies the number of consecutive SCHIBs starting at *vsch1* that are to be dumped. Specify END to dump all SCHIBs from *vsch1* to the highest number subchannel. END is the default.

#### \**dumpid*

identifies a 1- to 97-character string that is to be assigned to the dump. If specified, *dumpid* must be the last operand you enter on the command line.

## Usage Notes

1. If you specify a list of SCHIBS and one is not found, the DUMP command terminates at the first SCHIB found. For example, if you enter `dump SCHIB 0022 0033 0055 0011` and subchannel 0055 is not found, only SCHIB 0022 and 0033 will be displayed. If you specify a range of SCHIBs, the DUMP command terminates if the first SCHIB in the range is not found. Otherwise, all those SCHIBs that were found are dumped.
2. You can use the QUERY VIRTUAL command to determine the subchannel number associated with a virtual device.
3. The response to this command is a header line followed by a line of information for each SCHIB that you specify. (The header line identifies, column by column, the SCHIB information that follows it. See “Responses” below.)
4. For more information, see [“Usage Notes” on page 395](#).

## Responses

### Response 1: When EXP is not specified

```
SCHIB DEV INT-PARM ISC FLG LP PNO (Con't)
nnnn nnnn nnnnnnnnn n nn nn nn

LPU PI MBI PO PA CHPID0-3 CHPID4-7
nn nn nnnn nn nn nnnnnnnnn nnnnnnnnn
```

If an interrupt is pending for the virtual subchannel, the following additional line describes the pending interrupt:

```
KEY ECC FPI ZN SCTL CCW-ADDR DS SS CNT EXT-LOG
n n n n nnnn nnnnnnnnn nn nn nnnn nnnnnnnnn
```

### Response 2: When EXP is specified

```
SCHIB DEV INT-PARM ISC FLG MBI FLG2 MBA
nnnn nnnn nnnnnnnnn n nn nnnn nn nnnnnnnnnnnnnnnnn
```

### Response 3: When the command completes successfully

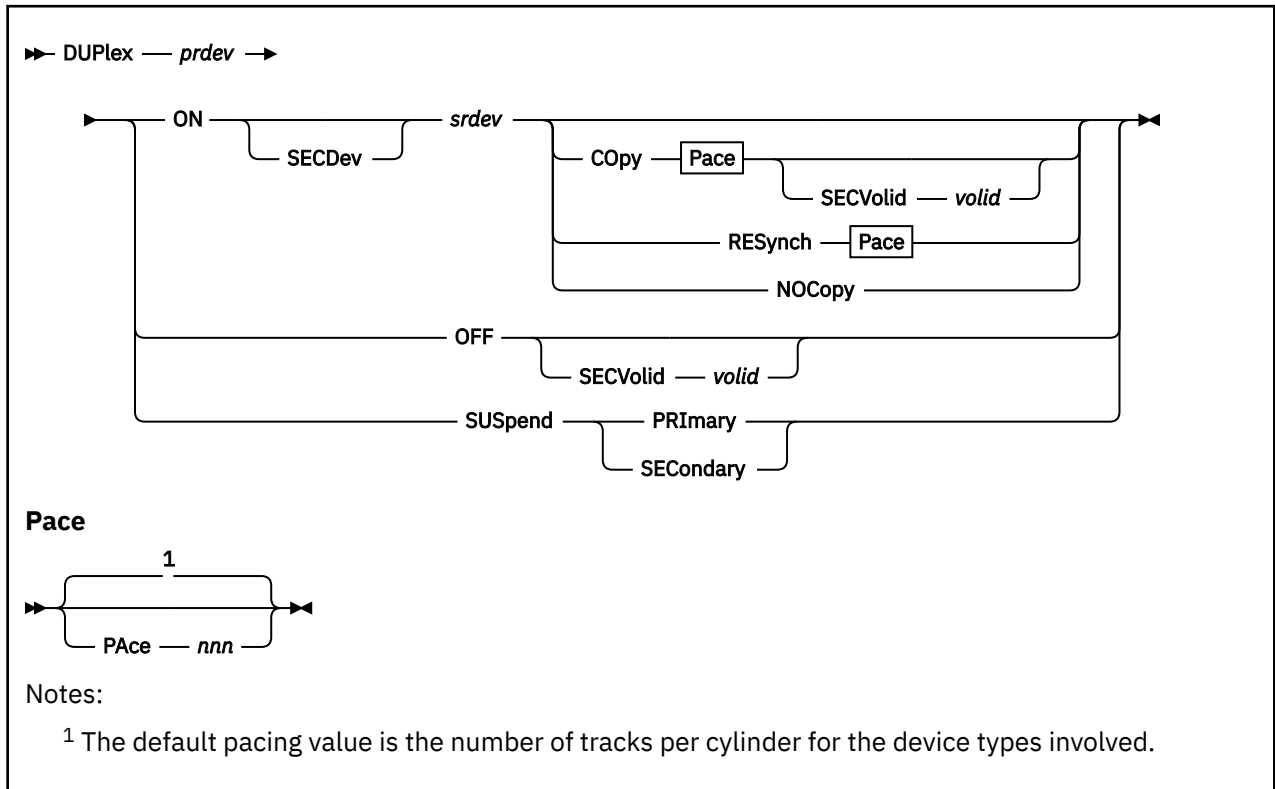
```
Command complete
```

Response header lines contain headings from the list below, a brief description is included.

Heading	Description
INT-PARM	The interrupt parameter
ISC	The interrupt subclass
FLG	The flag field
LP	The logical path mask
PNO	The path-not-operational mask
LPU	The last-path-used mask
PI	The path-installed mask
MBI	The measurement block index.
	<b>Note:</b> This will be ---- when format-1 measurement blocks are being used.
PO	The path-operational mask
PA	The path-available mask
CHPID	The channel path identifiers

<b>Heading</b>	<b>Description</b>
FLG2	The flag field (PMCW word 6, bits 24–31)
MBA	The measurement block address
KEY	The SCSW protect key
ECC	The logout condition code bits
FPI	The format, prefetch, and initial response bits
ZN	The initial status condition
SCTL	The function and activity control bytes
CCW-ADDR	The SCSW CCW address
DS	The SCSW device status bits
SS	The subchannel status bits
CNT	The SCSW residual CCW data count
EXT-LOG	The extended status/time word

# DUPLEX



## Authorization

Privilege Class: B

## Purpose

Use the DUPLEX command to perform duplex control operations (such as establishing or discontinuing a duplex pair) on devices attached to 3990 Model 3 and 3990 Model 6 subsystems.

This duplexing of devices makes use of the Dual Copy function, which allows two devices attached to the same 3990 Model 3 or 6 storage subsystem to operate in duplex mode. All data written to the primary device is automatically written to the specified secondary (copy) device. Thus, the host is relieved of performing this task when a duplicate copy of one or more DASD is desired.

## Operands

### prdev

is the real device number of the primary volume in the duplex pair.

### ON srdev

#### ON SECDev srdev

ON indicates that a duplex pair is to be established or re-established with the volumes specified.

SECDev is an optional keyword which may be used to clarify the fact that the real device number specified is the secondary device of a duplex pair.

The srdev is the real device number of the secondary volume which will be in the duplex pair when the command completes. This real device number specifies either:

- The secondary device of the duplex pair that is to be established or re-established after a SUSPEND



- A new secondary device to be used in establishing a new duplex pair by replacing the secondary device of an existing duplex pair.

**COPY**

indicates that synchronization of the contents of the two volumes by copying the entire contents of the primary volume to the secondary volume is to be performed to establish or re-establish the duplex pair. This is the default. See Usage Note 1.

**RESynch**

indicates that synchronization of the contents of the two volumes is to be performed to re-establish the duplex pair. Synchronization is accomplished by copying only modified tracks from the primary volume to the secondary volume. If the storage controller does not have a modified track map, the entire volume will be copied. See Usage Note 1.

**NOCopy**

indicates that synchronization of the contents of the two volumes is not to be performed to establish the duplex pair. See Usage Note 1.

**PAce *nnn***

indicates the rate at which synchronization of the two volumes is to take place if COPY or RESYNCH is specified. The value of *nnn* specifies the number of tracks to be copied by the control unit before allowing host access to the device for other operations. After allowing an interval for host access, the copying continues for another *nnn* tracks. This operation continues until the entire volume has been copied.

The value of *nnn* can be any number from 0 to 255. A value of 0 indicates that no pacing is to be performed. In other words, all tracks are to be copied without interruption, with the device being unavailable to all hosts.

If a pacing value is not specified for COPY or RESYNCH, the default pacing value is the number of tracks per cylinder for the device types involved. For example, the number of tracks per cylinder for 3380 devices is 15. Therefore, 15 would be the default for *nnn* of a 3380.

**Note:** Any volume that requires high availability (like a SYSRES) should use the smallest PAce value, which is 1.



**Attention:** Specifying PAce 0 can cause the primary volume of the duplex pair to be unavailable to **all** attached hosts for up to 10 to 20 minutes or more, depending on the size of the DASD. It may also severely disrupt ongoing operations for some, or all, hosts, depending on the data residing on the DASD and its use by the attached hosts.

You should use the default value. However, any value other than 0 will work, unless host missing-interruption intervals have been reduced to the 1- to 3-second range at your installation, in which case values above 60 should be avoided.

**SECVolid *valid***

specifies a new volume identification label for:

- The failed secondary volume. This occurs only when DUPLEX ON is issued to replace the failed volume of an existing duplex pair.
- The secondary volume of the duplex pair after duplexing is ended.

The *valid* is specified as a 1- to 6-character name. See Usage Note 6.

**SUSpend PRImary**

**SUSpend SECOndary**

indicates that the duplex pair identified by *prdev* is to be forced into suspended duplex state. The specified volume, either PRIMARY or SECONDARY, is marked as the suspended volume, and the other volume becomes the only operational volume of the duplex pair. See Usage Note 17.

**OFF**

indicates that the duplex relationship between the duplex pair specified by *prdev* is to be ended.

## Usage Notes

1. When using the DUPLEX ON command

- a. To establish a duplex pair, enter:

```
DUPlex prdev ON [SECDev] srdev [COPY] [PAce nnn]
```

or

```
DUPlex prdev ON [SECDev] srdev NOCOPY
```

**Note:** If NOCOPY is specified, the volumes must already be identical.

- b. To re-establish a suspended duplex pair, enter:

```
DUPlex prdev ON [SECDev] srdev [COPY] [PAce nnn]
```

or

```
DUPlex prdev ON [SECDev] srdev RESynch [PAce nnn]
```


- c. To replace the secondary volume of an existing suspended duplex pair, enter:

```
DUPlex prdev ON [SECDev] srdev
```

```
[COPY] [PAce nnn] [SECVolid volid]
```

***srdev***

is a new secondary volume to be used in establishing a new duplex pair.

2. This command requires that both the primary and the secondary volumes be in the CP I/O configuration.
3. Caching must be inactive for simplex devices being established as a duplex pair or joining an existing suspended duplex pair.
4. When re-establishing a suspended duplex pair, the pair must not have pinned data.
5. The PACE value is not allowed if NOCOPY is specified.
6. The volume identification label of the primary volume will be copied onto the secondary volume during the establishing of a new duplex pair, unless the NOCOPY option is specified. The SECVOLID option allows the primary and secondary volumes to have different identification labels after replacing the secondary volume of an existing duplex pair or ending a duplex pair.
7.  **Attention:** To assure that the secondary is a true copy of the primary when issuing DUPLEX OFF, the pair must not be pending or suspended and DASD fast write must be made inactive prior to issuing DUPLEX OFF.
8. The device identified by *srdev* must have a capacity that is equal to or greater than the capacity of the device identified by *prdev*.
9. DUPLEX ON requires that both cache and NVS be available to the subsystem.
10. When establishing a duplex pair where more than one host is sharing the environment, the device that is to be the secondary volume in the duplex pair must be varied offline to other hosts. For more information on [ungrouping storage paths](#), see *z/VM: System Operation*.
11. When establishing a new duplex pair, *prdev* and *srdev* must identify devices that are not currently in a duplex pair.
12. When re-establishing a duplex pair, *prdev* and *srdev* must be the primary and secondary volumes, respectively, of an established duplex pair.
13. When replacing the secondary volume of an existing duplex pair, *srdev* must identify a device that is not part of a duplex pair and *prdev* must identify an established duplex pair.

14. When re-establishing a duplex pair, or when replacing the secondary volume of an existing duplex pair, the duplex pair identified by *prdev* must be in suspended duplex state or the command will not complete successfully.
15. For DUPLEX ON, the secondary volume identified by *srdev* may be an offline device.
16. For DUPLEX SUSPEND, if the duplex pair is in suspended duplex state, the command will not complete successfully.
17. If DUPLEX SUSPEND PRIMARY is issued, the storage control will swap the primary and secondary device addresses as seen by the host. This allows a functioning device to continue to be represented by the primary device address. To determine the physical device address, see [“QUERY DASD” on page 867](#) and in particular Response 7 for that command.
  - If DUPLEX ON SECDEV is issued to re-establish the duplex pair, the addresses **will** be swapped back to their original state as part of the operation.
  - If DUPLEX OFF is issued to end the duplex pair, the addresses **will not** be swapped back to their original state.
  - If DUPLEX ON SECDEV is issued to replace the suspended secondary with another device, the addresses **will not** be swapped back to their original state. The freed device will be in simplex state on the physical address to which it is currently assigned. However, if the address of the new device, as seen by the host, is the same as the physical address of the suspended device, then the address of the new device will be swapped with the address of the freed device. This is possible if the device has been put in suspended duplex state several times and address swaps have occurred.
18. Once a duplex pair has been established, the caching status and the DASD fast write status of the primary volume becomes the caching and DASD fast write status of the duplex pair.
19. When DUPLEX OFF is used to discontinue a duplex pair, the caching status and the DASD fast write status of the former primary volume remains the same as the caching and DASD fast write status of the discontinued duplex pair. For the former secondary volume, the caching status is made active and the DASD fast write status is made inactive.
20. A volume attached to the system can only be duplexed with a free volume. The volume that is attached to the system must be the primary of the duplex pair.
21. A volume that is dedicated to a guest virtual machine can only be duplexed with a free volume or another volume dedicated to the same user. The volume that is dedicated to the guest must be the primary of the duplex pair.

The ATTACH command can be used to attach the secondary volume to the guest virtual machine after the duplex pair has been established.



**Attention:** Some guests, MVS for example, require that both the primary and the secondary be attached.

22. The primary device of the duplex pair can be a free volume only when the secondary device of the duplex pair is a free volume.
23. When a duplex pair has been established by CP, the guest will not be allowed to fail the primary if permanent errors occur on valid-defined minidisks. The recovery action will have to be done manually with the DUPLEX command.
24. The primary and secondary volumes should be on different DASD strings. This provides greater data availability than duplex pairs on the same DASD string.
25. This command may produce delayed responses.

## Responses

Response 1:

Response 1 has three types of responses. When establishing or re-establishing a duplex pair, one or more of the following response types will occur:

- Immediate Response

## DUPLEX

- Delayed Response
- Response to System Operator

*Immediate Response:* If a duplex pair is established or re-established immediately, this response will be displayed:

```
Duplex pair (prdev,srdev) - active
```

where:

**prdev**

is the real device number of the primary volume in the duplex pair.

**srdev**

is the real device number of the secondary volume in the duplex pair.

*Delayed Response:* If synchronization of the contents of the two volumes is in progress, this response will be displayed:

```
Command started:  
DUPLEX prdev ON srdev {COPY } PACE nnn [SECVolid valid]  
                {RESYNCH}
```

where:

**prdev**

is the real device number of the primary volume in the duplex pair.

**srdev**

is the real device number of the secondary volume in the duplex pair.

**nnn**

is the copy pace value.

**valid**

is the volume identification label for the former secondary volume after DUPLEX ON is issued to replace the secondary volume of an existing duplex pair.

When the duplex pair is established or re-established, the following response will be displayed.

```
Duplex pair (prdev,srdev) - active
```

Sometimes a cache storage subsystem will not notify CP when it has established or re-established a duplex pair. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing is ended after generating the following response:

```
Command results lost:  
DUPLEX prdev ON srdev {COPY } PACE nnn [SECVolid valid]  
                {RESYNCH}
```

Then the QUERY DUPLEX command can be issued to determine the status of the duplex pair.

Errors can occur during the asynchronous processing of the DUPLEX command. If these errors prevent the successful completion of the command, the following response will be displayed:

```
Command failed:  
DUPLEX prdev ON srdev {COPY } PACE nnn [SECVolid valid]  
                {RESYNCH}
```

*Response to System Operator:* If the originator of the DUPLEX ON command is not the system operator, this response will be issued to the system operator when the duplex pair has been established or re-established.

```
Duplex pair (prdev,srdev) was activated by userid.
```

where:

***prdev***

is the real device number of the primary volume in the duplex pair.

***srdev***

is the real device number of the secondary volume in the duplex pair.

***userid***

is the ID of the user who issued the command.

Response 2:

```
Duplex pair (prdev,srdev) - suspended
```

This response is displayed when DUPLEX SUSPEND command is issued to place a duplex pair in the suspended duplex state.

where:

***prdev***

is the real device number of the primary volume in the duplex pair.

***srdev***

is the real device number of the secondary volume in the duplex pair.

If the originator of the command is not the system operator, the following response will be issued to the system operator.

```
Duplex pair (prdev,srdev) was suspended by userid.
```

where:

***prdev***

is the real device number of the primary volume in the duplex pair.

***srdev***

is the real device number of the secondary volume in the duplex pair.

***userid***

is the ID of the user who issued the command.

Response 3:

```
Devices prdev and srdev are no longer duplexed.
```

This response is displayed when DUPLEX OFF is issued to discontinue the duplex relationship between two devices.

where:

***prdev***

is the real device number of the volume that was the primary volume of the duplex pair.

***srdev***

is the real device number of the volume that was the secondary volume of the duplex pair.

If the originator of the command is not the system operator, the following response will be issued to the system operator.

```
Duplex relationship between prdev and srdev was discontinued by userid.
```

where:

***prdev***

is the real device number of the volume that was the primary volume of the duplex pair.

***srdev***

is the real device number of the volume that was the secondary volume of the duplex pair.

## DUPLEX

### **userid**

is the ID of the user who issued the command.

Response 4:

```
srdev Volume label is now 'XXXXXX'.
```

This response is displayed when DUPLEX ON or DUPLEX OFF is issued to specify a new volume identification label for the former secondary volume of the duplex pair.

where:

### **srdev**

is the real device number of the volume that was the secondary volume of the duplex pair.

### **XXXXXX**

is volume identification label.

If the originator of the command is not the system operator, the following response will be issued to the system operator.

```
srdev Volume label was changed by userid and is now 'XXXXXX'.
```

where:

### **srdev**

is the real device number of the volume that was the secondary volume of the duplex pair.

### **userid**

is the ID of the user who issued the command.

### **XXXXXX**

is volume identification label.

Response 5:

```
srdev Changing of volume label has failed.
```

This response is displayed when the DUPLEX ON or DUPLEX OFF command cannot specify a new volume identification label for the former secondary volume of the duplex pair.

where:

### **srdev**

is the real device number of the volume that was the secondary volume of the duplex pair.

Response 6:

```
Duplex pair (prdev,srdev) - not established  
because cache and/or non-volatile storage is not available.
```

This response is generated when DUPLEX ON is issued and cache and/or NVS was not available to the subsystem.

where:

### **prdev**

is the real device number of the volume specified as the primary volume of the duplex pair.

### **srdev**

is the real device number of the volume specified as the secondary volume of the duplex pair.

Response 7:

```
Duplex pair (prdev,srdev) - not established  
because caching is active for one or both devices.
```

This response is generated when DUPLEX ON is issued and caching was active for one or both of the devices.

where:

**prdev**

is the real device number of the volume specified as the primary volume of the duplex pair.

**srdev**

is the real device number of the volume specified as the secondary volume of the duplex pair.

Response 8:

```
Duplex pair (prdev,srdev) - not re-established
because the duplex pair is not suspended.
```

This response is generated when DUPLEX ON is issued to re-establish a duplex pair or to replace the secondary, and the duplex pair identified by *prdev* was not in suspended duplex state.

where:

**prdev**

is the real device number of the volume specified as the primary volume of the duplex pair.

**srdev**

is the real device number of the volume specified as the secondary volume of the duplex pair.

Response 9:

```
Duplex pair (prdev,srdev) - already suspended
```

This response is generated when DUPLEX SUSPEND is issued and the duplex pair was already in suspended duplex state.

where:

**prdev**

is the real device number of the volume that is the primary volume of the suspended duplex pair.

**srdev**

is the real device number of the volume that is the secondary volume of the suspended duplex pair.

Response 10:

```
Devices prdev and srdev cannot be duplexed.
```

This response is generated when DUPLEX ON is issued and the duplex pair cannot be created for one of the following reasons:

- The devices are not configured into the same storage subsystem.
- The devices do not have the same capacity.
- One or both of the devices are already duplexed with other devices.
- A volume attached to the system or to a guest was to be duplexed and either was not specified as the primary device, or the specified secondary device was not a free volume.
- The primary device is free and the secondary device is attached to the system or to a guest.

where:

**prdev**

is the real device number of the volume that was specified as the primary volume in the DUPLEX command.

**srdev**

is the real device number of the volume that was specified as the secondary volume in the DUPLEX command.

Response 11:

## DUPLEX

*prdev* The associated secondary volume in the duplex pair is not online to VM.

This response is displayed when the device number of the primary volume is specified on the DUPLEX SUSPEND or DUPLEX OFF command, and the device number of the secondary volume cannot be located.

where:

***prdev***

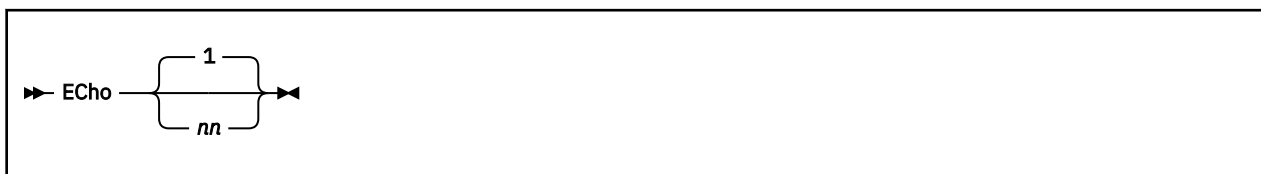
is the real device number of the primary volume in the duplex pair.

### Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP021E A real device number was not supplied or it is invalid.
- HCP023E VOLID missing or invalid
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*
- HCP1111E Device *rdev* is not the primary volume of a duplex pair.
- HCP1112E Device *srdev* is not the associated secondary volume with device *prdev*.
- HCP2227E Device *rdev* is a volume of a PPRC pair.



## ECHO



### Authorization

Privilege Class: G

### Purpose

Use ECHO to put your terminal in the echo environment.

**Note:** In the echo environment, any input line you enter is transmitted back to your terminal *nn* times.

### Operands

**1**

***nn***

identifies the number of times that the line is repeated. The default is 1. An entry that is not valid—that is, one that is less than 1 or greater than 99 or contains nonnumeric characters is treated as 1.

### Usage Notes

1. After the line has been repeated the specified number of times, CP issues another read to the display so you can enter another data line.
2. No line editing is done; the output line is the same as the input line and may contain any of the logical line editing symbols.
3. The responses and redisplayed input lines are sent to the terminal only. They are not passed to the virtual machine.

### Responses

Response 1:

```
ECHO entered; to terminate test, type end.
```

is displayed after you have issued the ECHO command to indicate that your virtual machine has entered the echo environment.

Response 2:

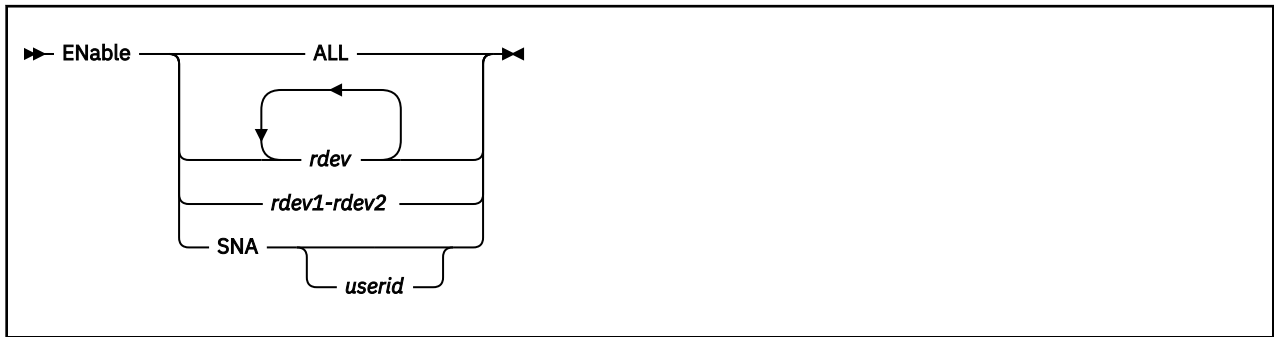
```
Enter line.
```

is displayed to prompt you to enter an input line. If you enter the reply END, you exit from the ECHO environment.

### Messages

- HCP002E Invalid operand - *operand*

## ENABLE (Device)



### Authorization

Privilege Class: A, B, F

### Purpose

Use ENABLE (device) to enable the following devices to access the host system:

- Disabled local 3270 display devices
- EP (Emulator Program) controlled lines in the 3745
- SNA/CCS terminals.

### Operands

#### ALL

enables all previously disabled or non-enabled locally attached 3270 display devices, including 328x printers, EP (Emulator Program) controlled lines in the 3745, and all SNA/CCS terminals.

#### *rdev*

#### *rdev1-rdev2*

is the device number, a list of device numbers, or a range of device numbers of display stations or lines that you want to enable. If you specify a list of device numbers, separate them by one or more blanks.

#### **SNA** *userid*

enables communication with the SNA/CCS terminals owned by the VTAM service machine whose user ID is specified. If no user ID is specified, communication is enabled with all SNA/CCS terminals for all VTAM service machines.

### Usage Notes

1. On a single command line you may combine device numbers, lists of device numbers, and ranges of device numbers.
2. Do not attempt to enable display devices that are not supported. Use unsupported console devices as dedicated devices attached to virtual machines.
3. If a display station or line is already enabled, it is not affected by the ENABLE (device) command.

### Responses

Response 1:

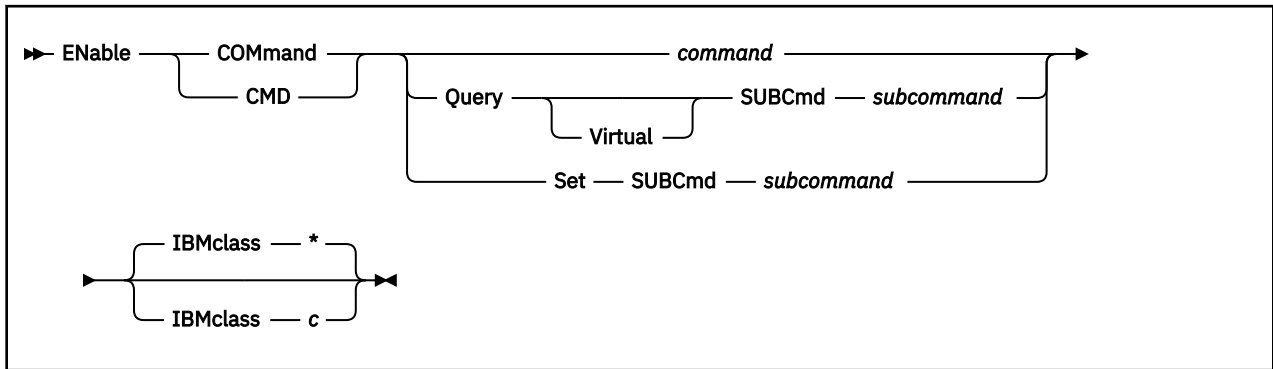
```
Command complete
```

is issued when the command completes successfully. No error indication is provided if no displays or lines are available to be enabled.

## Messages

- HCP002E Invalid operand - *operand*
- HCP006E Invalid device type - {*rdev* | *vdev* | *ldev*}
- HCP009E Invalid range - *range*
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev* | *vdev* | *ldev*} does not exist
- HCP046E *type rdev* offline
- HCP053E [XAUTOLOG failed for *userid*:] *userid* | *value* not in CP directory
- HCP140E *type* {*rdev* | *ldev*} attached to *userid*
- HCP263E Too many operands were supplied for this command.

## ENABLE COMMAND / CMD



### Authorization

Privilege Class: A

### Purpose

Use ENABLE COMMAND or CMD to permit CP to process requests for the specified CP command.

You can also permit processing of CP commands during initialization using the ENABLE COMMAND or CMD configuration file statements. For more information, see [ENABLE COMMAND / CMD Statement in z/VM: CP Planning and Administration](#).

### Operands

#### *command*

is the name of the command that you are enabling. The variable *command* is a 1- to 12-character alphanumeric string.

#### **Query SUBCmd *subcommand***

tells CP the name of the CP QUERY subcommand that you are enabling. The variable *subcommand* is a 1- to 12-character alphanumeric string.

#### **Query Virtual SUBCmd *subcommand***

tells CP the name of the CP QUERY VIRTUAL subcommand that you are enabling. The variable *subcommand* is a 1- to 12-character alphanumeric string.

#### **Set SUBCmd *subcommand***

tells CP the name of the CP SET subcommand that you are enabling. The variable *subcommand* is a 1- to 12-character alphanumeric string.

#### **IBMclass \***

tells CP to enable all versions of the specified command or subcommand. If omitted, IBMCLASS \* is the default.

#### **IBMclass c**

tells CP to enable a specific version of the specified command or subcommand. The variable *c* can be any 1 of the following:

- A** this is a system-control command to be used by the primary system operator.
- B** this is a command for operational control of real devices.
- C** this is a command to alter system storage.

- D** this is a command for system-wide control of spool files.
- E** this is a command to examine system storage.
- F** this is a command for service control of real devices.
- G** this is a general-use command used to control the functions of a virtual machine.
- O** (zero) this command has no specific IBM class assigned.

## Usage Notes

1. To load the command processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD” on page 111](#)).
2. To activate a CP command while defining it, use the ENABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD” on page 143](#)).
3. Both an alias and its base command must be enabled for the alias to function. If the command you want to enable is defined as an alias, then you must also enable the base command for the alias to function.
4. To deactivate a CP command:
  - While defining it, use the DISABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD” on page 143](#)).
  - After defining it, use the DISABLE COMMAND or CMD command (see [“DISABLE COMMAND / CMD” on page 313](#)).
5. To change the definition of an existing CP command, use the MODIFY COMMAND or CMD command (see [“MODIFY COMMAND / CMD” on page 683](#)).
6. To obtain information such as the enable/disable status and information as to whether a command is an alias or not, use the QUERY CPCMDS command (see [“QUERY CPCMDS” on page 817](#)).
7. To display the address of the CP command table entry block, the current IBM class, and the current privilege class for a specified CP command, use the LOCATE CMDBK command (see [“LOCATE CMDBK” on page 603](#)).
8. To remove the command processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD” on page 117](#)).

## Responses

### Response 1:

To activate the CP SHUTDOWN command (after a prior DISABLE COMMAND command or configuration file statement had deactivated it), enter the following:

```
enable command shutdown
Ready;
```

### Response 2:

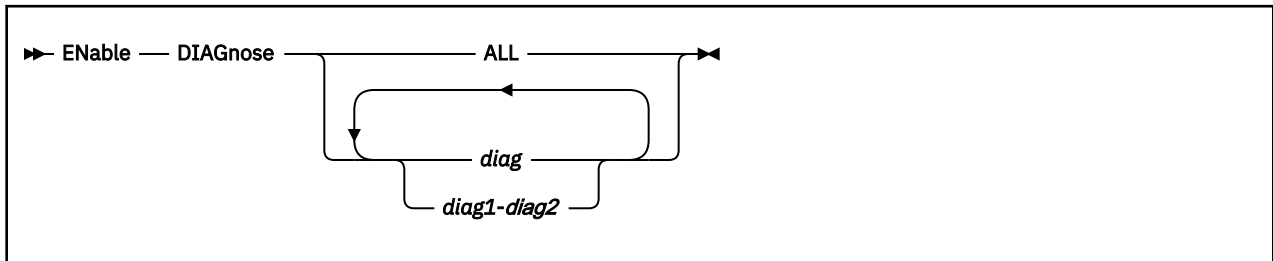
To activate the class <ANY> version of the CP SET PRIVCLASS command (after a prior DISABLE COMMAND command or configuration file statement had deactivated it), enter the following:

```
enable command set subcmd privclass ibmclass 0
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid command name - *command*
- HCP6706E Invalid IBM class - *c*
- HCP6706E Invalid privilege classes - *classes*
- HCP6706E Invalid string - *string*
- HCP6706E Invalid subcommand name - *subcommand*
- HCP6840E You specified QUERY VIRTUAL without SUBCMD
- HCP6840E You specified SUBCMD without SET or QUERY
- HCP6841E You did not specify an IBM class.
- HCP6842E Cannot find existing CP command
- HCP9300E A system soft abend *code* has occurred on CPU *nnnn* for user *userid*.

## ENABLE DIAGNOSE



### Authorization

Privilege Class: A

### Purpose

Use ENABLE DIAGNOSE to permit CP to process requests for one or more locally-developed DIAGNOSE codes.

You can also permit processing of locally-developed DIAGNOSE codes during initialization using the ENABLE DIAGNOSE configuration file statement. For more information, see [ENABLE DIAGNOSE Statement](#) in *z/VM: CP Planning and Administration*.

### Operands

#### ALL

tells CP to enable all existing DIAGNOSE codes.

**Note:** This operand enables all DIAGNOSE codes: the locally-defined ones, the IBM-supplied ones, and any supplied by third-party software vendors.

#### *diag*

#### *diag1-diag2*

is the number of the DIAGNOSE code that you are enabling. Each *diag* must be a hexadecimal number between X'0000' and X'03FC' and must be a multiple of 4. You can specify a single DIAGNOSE code, a range of DIAGNOSE codes, or any combination thereof.

### Usage Notes

1. To define a new DIAGNOSE code, use the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE”](#) on page 172).
2. To load the DIAGNOSE processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD”](#) on page 111).
3. If you do not specify the ENABLE operand, a new DIAGNOSE code is initially in a disabled state after being defined. CP treats disabled DIAGNOSE codes as if they were never defined. If you try to use a disabled DIAGNOSE code in a program, CP will give you a program check specification exception.
4. To change the definition of an existing DIAGNOSE code after initialization, use the MODIFY DIAGNOSE command (see [“MODIFY DIAGNOSE”](#) on page 697).
5. To display information about a DIAGNOSE code (status, entry point name, and privilege class), use the QUERY DIAGNOSE command (see [“QUERY DIAGNOSE”](#) on page 885).
6. To display the address of the CP DIAGNOSE code table block for a DIAGNOSE code, use the LOCATE DGNBK command (see [“LOCATE DGNBK”](#) on page 608).
7. To deactivate a DIAGNOSE code:

## ENABLE DIAGNOSE

- While defining it, use the DISABLE operand of the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE”](#) on page 172).
  - After defining it, use the DISABLE DIAGNOSE command (see [“DISABLE DIAGNOSE”](#) on page 316).
8. To remove the DIAGNOSE processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117).
  9. Many external security managers (ESMs) do not support DIAGNOSE codes above X'03FC'. For this reason, CP does not support DIAGNOSE codes above X'03FC'. The DIAGNOSE codes between X'0000' and X'03FC' are divided as follows:
    - X'0000' to X'00FC'**  
Reserved for IBM use
    - X'0100' to X'01FC'**  
Reserved for customer use
    - X'0200' to X'03FC'**  
Reserved for IBM use.
  10. For more information about user-defined DIAGNOSE codes, see [Defining and Modifying Commands and Diagnose Codes](#) in *z/VM: CP Exit Customization*.

## Responses

Response 1 (Enable One):

To enable DIAGNOSE code X'100', enter the following:

```
enable diagnose 100
Ready;
```

Response 2 (Enable All):

To enable all DIAGNOSE codes on your system, enter the following:

```
enable diagnose all
Ready;
```

Response 3 (Enable All Local):

To enable all locally-defined DIAGNOSE codes, enter the following:

```
enable diagnose 100-1fc
Ready;
```

Response 4 (Enable Almost All Local):

To enable all locally-defined DIAGNOSE codes except DIAGNOSE code X'180', enter the following:

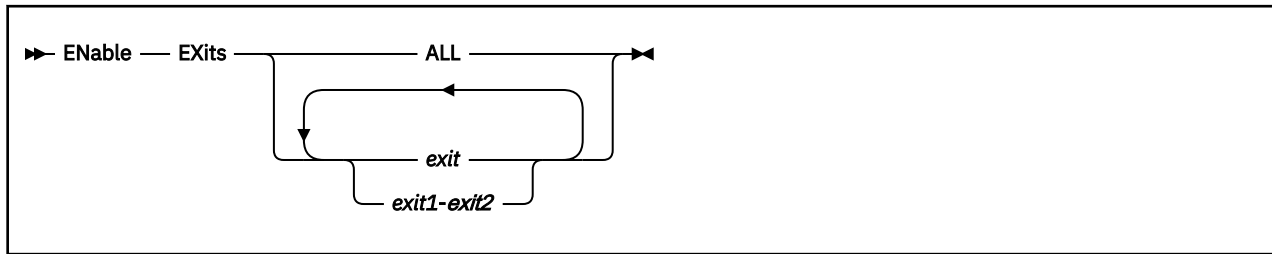
```
enable diagnose 100-17c 184-1fc
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP2751E Diagnose code *diag* does not exist
- HCP2751E Diagnose codes *diag1-dia2* do not exist
- HCP6704E Missing token at end of line
- HCP6706E Invalid Diagnose code - *diag*



## ENABLE EXITS



### Authorization

Privilege Class: A

### Purpose

Use ENABLE EXITS to permit CP to call all entry points and external symbols associated with one or more exit points.

You can also permit CP to call the entry points and external symbols associated with an exit point during initialization using the ENABLE EXITS configuration file statement. For more information, see [ENABLE EXITS Statement](#) in *z/VM: CP Planning and Administration*.

### Operands

#### ALL

tells CP to enable all existing CP exit points.

#### exit

#### exit1-exit2

is the number of the exit point (or exit points) that you want CP to start using. Each *exit* must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single exit point number, a range of exit point numbers, or any combination thereof.

### Usage Notes

1. To load the exit point code into the system execution space, use the CPXLOAD command (see [“CPXLOAD”](#) on page 111).
2. To associate one or more entry points or external symbols with a specific exit point and to enable or disable that exit point, use the ASSOCIATE EXIT command (see [“ASSOCIATE EXIT”](#) on page 20). You can also use the ASSOCIATE EXIT command to change the entry points and external symbols associated with a specific entry point. If the exit point is enabled when you change the list of entry points and external symbols, CP checks to see if that exit point is currently processing an exit request. If the exit point is not processing any requests, CP immediately updates the list with your changes. If the exit point is processing a request, CP waits until the current request is completed and then updates the list with your changes before processing any new exit requests.
3. If the list of entry points and external symbols associated with this exit point contain any entry points or external symbols that CP does not know about, CP just ignores them and continues normal processing. That is, CP will continue to process the other members of the list associated with this exit point. CP does not ignore an exit point because it cannot find one entry point or external symbol in the list. CP only ignores an exit point if it cannot find all the entry points and external symbols in the list.
4. To display whether there are any unknown entry points or external symbols associated with an exit point, use the QUERY UNRESOLVED command (see [“QUERY UNRESOLVED”](#) on page 1250).

5. To display status and usage statistics information about a specific exit point, use the QUERY EXITS command (see [“QUERY EXITS”](#) on page 907).
6. To display the address of the CP exit block for a specific exit point, use the LOCATE XITBK command (see [“LOCATE XITBK”](#) on page 649).
7. To display the address of the CP indirect call locator block for a specific exit point, use the LOCATE ICLBK command (see [“LOCATE ICLBK”](#) on page 614).
8. To change the definition of an existing dynamic exit point, or to remove the exit point from the system, use the MODIFY EXIT command (see [“MODIFY EXIT”](#) on page 700).
9. To stop CP from calling all the entry points and external symbols associated with one or more exit points after defining those exit points, use the DISABLE EXITS command (see [“DISABLE EXITS”](#) on page 319).
10. To remove the exit point code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117).
11. For more information about user-defined exit points, see [z/VM: CP Exit Customization](#).

## Responses

Response 1:

To have CP start using the entry points and external symbols associated with CP Exit 99, enter the following:

```
enable exits 99
Ready;
```

Response 2:

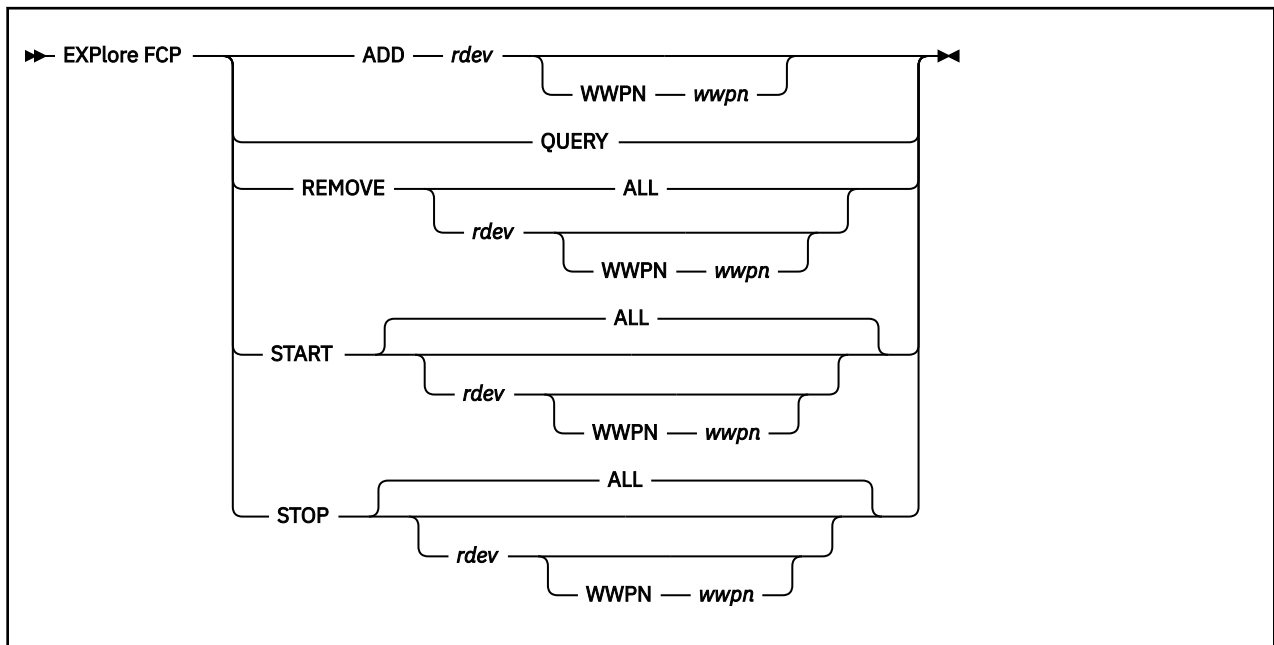
To have CP start using the entry points and external symbols associated with all CP exit points, enter the following:

```
enable exits all
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP2752E Exit number *exit* does not exist
- HCP2752E Exit numbers do not exist
- HCP2752E Exit numbers *exit1-exit2* do not exist
- HCP6704E Missing token at end of line
- HCP6706E Invalid CP Exit number - *exit*

## EXPLORE FCP



### Authorization

Privilege Class: B

### Purpose

Use EXPLORE FCP to define a set of FCP subchannels and WWPN ports to be tested, to manage the list, and to start and stop the devices. This command can be used to activate the FCP subchannels and open the WWPN ports so that they are visible to the SAN fabric and can be verified on storage array consoles.

### Operands

#### ADD

adds an FCP subchannel and optionally a WWPN port to the CP-managed list of SCSI devices to be tested.

#### QUERY

displays the FCP subchannels and WWPN ports, along with their current activation status, from the list of SCSI devices to be tested.

#### REMOVE

removes an FCP subchannel and optionally a WWPN port from the CP-managed list of SCSI devices to be tested.

#### START

activates FCP subchannels and opens WWPN ports in the list of SCSI devices to be tested.

#### STOP

deactivates FCP subchannels and closes WWPN ports in the list of SCSI devices to be tested.

### Options

#### ALL

forces the related subcommand to process all elements in the list of SCSI devices to be tested.

#### rdev

specifies the real device number of the FCP subchannel.

### WWPN *wwpn*

specifies the world wide port name to be processed.

### Usage Notes

1. When the ADD subcommand is issued, the FCP device specified must be free and will be attached to SYSTEM. The FCP device remains attached to SYSTEM until it, and all WWPNs associated with it, are removed using the REMOVE subcommand.
2. In order to remove a device using the REMOVE subcommand, the device must first be deactivated using the STOP subcommand.

### Responses

Response 1:

If you issue EXPLORE FCP QUERY after adding a port to the list, you will see the following:

```
explore fcp query
FCP: 1C20 STATUS: INACTIVE
FCP: 1C20 WWPN: 5005076300CD04DA STATUS: INACTIVE
```

Response 2:

If you issue EXPLORE FCP QUERY after an EXPLORE FCP START command, you will see the following:

```
explore fcp query
FCP: 1C20 STATUS: ACTIVE
FCP: 1C20 WWPN: 5005076300CD04DA STATUS: ACTIVE
```

Response 3:

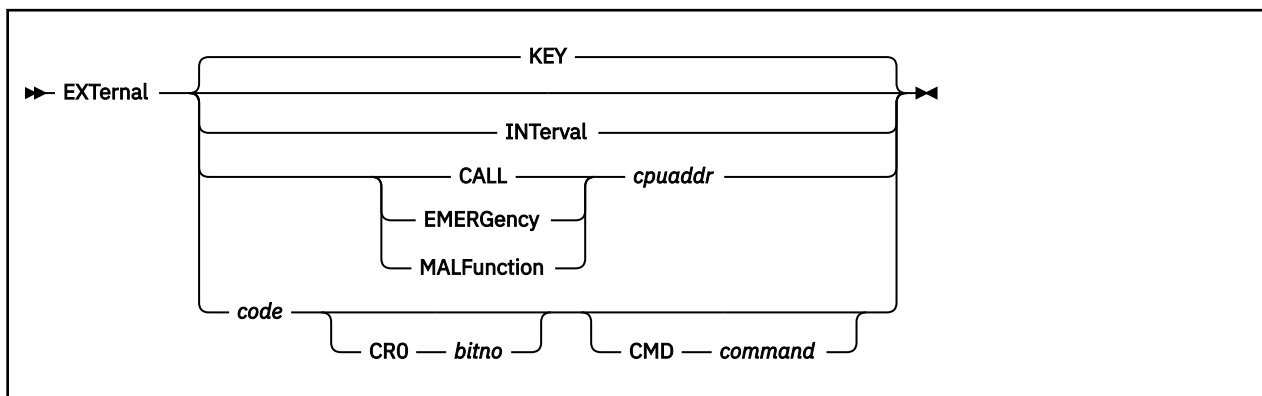
If you issue EXPLORE FCP QUERY after adding an FCP device without a WWPN to the list, you will see the following:

```
explore fcp query
FCP: 1C21 STATUS: INACTIVE
```

### Messages

- HCP8713E Duplicate port specified
- HCP8714E Port cannot be removed while it is active
- HCP8715E Specified port is not found
- HCP8715E Specified port is not active
- HCP8715E Specified port is already active
- HCP8716E Failed to activate device *rdev*
- HCP8716E Failed to deactivate device *rdev*
- HCP8717E Failed to open port *wwpn* on device *rdev*
- HCP8717E Failed to close port *wwpn* on device *rdev*

## EXTERNAL

**Authorization**

Privilege Class: G

**Purpose**

Use EXTERNAL to simulate external-interrupt-pending to your virtual machine.

**Note:** The virtual machine takes immediate control.

**Operands****KEY**

identifies the operator external interrupt key. KEY is the default.

**INTerval**

identifies the interval timer at location 80. You can specify INTERVAL only if the 370 Accommodation facility is active. Use the CP SET 370ACCOM ON command to activate the 370 Accommodation facility. When you specify INTERVAL, an interval timer interrupt is made pending.

**CALL**

identifies an external call SIGP signal from the specified virtual processor address and is accepted only when no external call is already pending.

**EMERgency**

identifies an emergency signal from the specified virtual processor address.

**MALFunction**

identifies a malfunction alert signal from the specified virtual processor address.

***cpuaddr***

identifies a virtual processor address from 0 to 3F (hexadecimal).

***code***

identifies an external interrupt code.

***CRO bitno***

indicates that a mask bit in CRO (control register 0) is to be applied to this interrupt. The operand *bitno* is the bit number in CRO that masks this interrupt; it is a decimal number from 16 to 31. For an ESA/390 virtual machine, specifying 16-31 refers to the masks in bits 16-31 of control register 0. For a z/Architecture mode virtual machine, specifying 16-31 refers to the corresponding masks that are in bits 48-63 of control register 0.

### **CMD command**

indicates that a CP command is to be issued when the interrupt is reflected. You must enter the command at the end of the command line; characters following CMD are interpreted as the CP command.

### **Usage Notes**

1. The BEGIN command is not required after you enter the EXTERNAL command.
2. Since there is no virtual processor address argument for code:
  - a. EXTERNAL 1200 cannot be an alternative syntax for EXTERNAL MALFUNCTION *cpuaddr*
  - b. EXTERNAL 1201 cannot be an alternative syntax for EXTERNAL EMERGENCY *cpuaddr*
  - c. EXTERNAL 1202 cannot be an alternative syntax for EXTERNAL CALL *cpuaddr*.

**Note:** This is true because of the architectural requirement that only 1 external call can be pending for a virtual processor at any one time.

A bit in CRO can be specified to mask this interrupt. Since the interrupt code does not imply a bit in CRO:

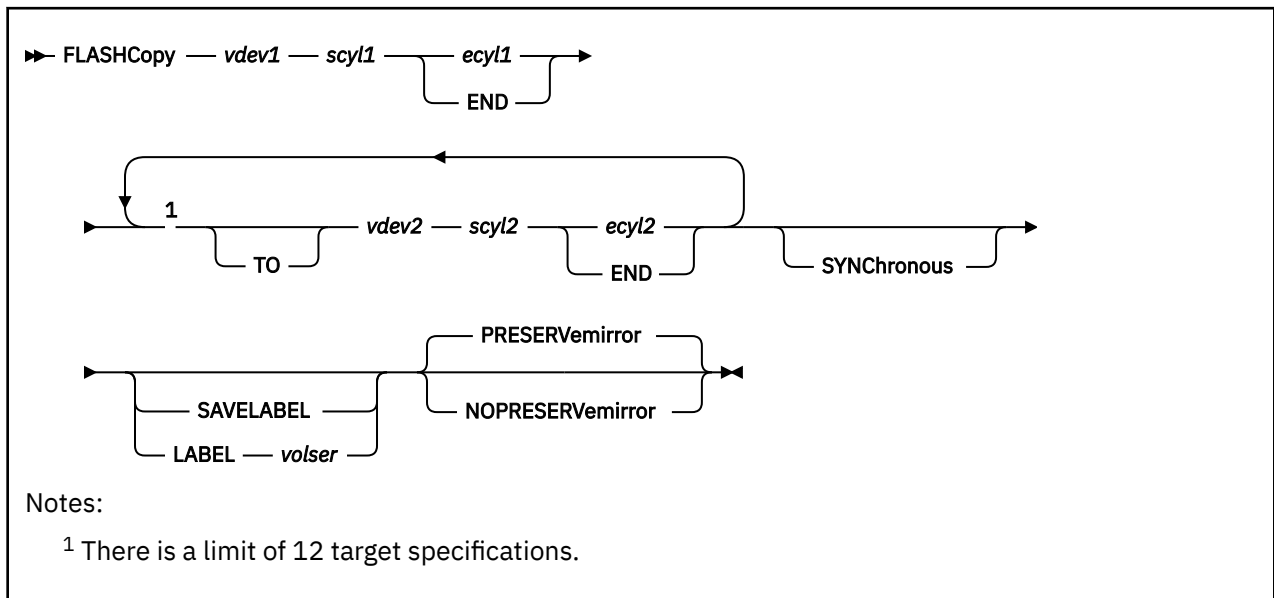
  - a. EXTERNAL 40 is not an alternative syntax for EXTERNAL KEY.
  - b. EXTERNAL 80 is not an alternative syntax for EXTERNAL INTERVAL.

**Note:** This is true because of the architectural requirement that only 1 interval timer interrupt can be pending for a virtual processor at any one time.
3. You can specify a CP command that you want to be processed after the interrupt is reflected to your virtual machine. Use this in system testing.

### **Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*
- HCP026E Operand missing or invalid

## FLASHCOPY



### Authorization

Privilege Class: B

### Purpose

Use FLASHCOPY to copy data from one virtual DASD to another.

### Operands

#### **vdev1**

is the address of the virtual DASD to be used as the source of the copy. The virtual DASD must be on a control unit that has the FLASHCOPY hardware feature. If a valid address is not specified, the command is rejected.

#### **scyl1**

is the first cylinder to be copied. If the whole virtual device is to be copied, this should be 0.

#### **ecyl1**

is the last cylinder to be copied. If the whole virtual device is to be copied, this should be END.

#### **TO**

#### **vdev2**

is the address of the virtual DASD to be used as the target of the copy. The virtual DASD must be on a control unit that has the FLASHCOPY hardware feature and must not be read only. If a valid address is not specified, the command is rejected. If multiple targets are entered, the same source device will be copied to each target device.

#### **scyl2**

is the first cylinder to be written. If the whole virtual device is to be written, this should be 0.

#### **ecyl2**

is the last cylinder to be written. If the whole virtual device is to be written, this may be END.

#### **SYNChronous**

(For FlashCopy Version 1 hardware) tells CP to process the command immediately and to wait until the hardware has accepted all parameters, all messages from the Enterprise Storage Server® (ESS) subsystem have been processed, and the FLASHCOPY command completes.

**SAVELABEL**

asks CP to preserve the volume label on *vdev2* virtual cylinder 0, head 0, record 3. This option attempts to read and verify the volume label from the target devices **before** the relationship is established. This means that a volume label should exist on the target before this command is issued. After the relationship is established, the same processing as for the LABEL option is performed to update the label.

**LABEL volser**

asks CP to change the volume label on *vdev2* virtual cylinder 0, head 0, record 3 to *volser*. This option reads and verifies the volume label on the target virtual cylinder 0, head 0, record 3 **after** the relationship has been established. This implies that the target must contain a valid label that was either copied from the source or not overwritten at all. Once the volume label has been verified, it is rewritten with the specified *volser* onto target cylinder 0, head 0, record 3.

**PRESERVEmirror**

requests that the Flashcopy operation must not change the target PPRC pair to a duplex-pending status.

**NOPRESERVEmirror**

requests that the Flashcopy operation can complete with no requirements on the PPRC status of the target devices.

**Usage Notes**

1. In order for this command to be successful, the source and target DASD must be correctly defined on control units that support the FLASHCOPY feature. If this is not the case, error message HCP332E is generated.
2. This command might produce delayed responses. Refer to [“Delayed Responses to CP Commands” on page 7](#) for more information.
3. Special care should be exercised when copying to cylinder zero of the target virtual device. If the target virtual device starts at real cylinder zero, the CP volume label will be overwritten. This could cause a duplicate volume to be created or a volume that cannot be mounted.
4. FLASHCOPY supports source-target pairs according to the DASD types of the source and target. [Table 18 on page 472](#) shows the source-target pairs FLASHCOPY supports. If you attempt to issue a FLASHCOPY command to an unsupported source-target DASD pair, you will receive message HCP6706E.

*Table 18. Source-target pairs supported by FLASHCOPY*

	<b>Target: Dedicated</b>	<b>Full-pack</b>	<b>Not full-pack</b>
<b>Source:</b>			
<b>Dedicated</b>	supported	supported	
<b>Full-pack</b>	supported	supported	
<b>Not full-pack</b>			supported

5. For FlashCopy Version 1 hardware, issuing the FLASHCOPY command to a minidisk will fail if the source and target minidisks occupy different physical extents with respect to their physical volumes. For example, minidisk 101 has a starting location of 101 and minidisk 102 has a starting location of 26. If the following command is issued:

```
FLASHCOPY 101 0 END 102 0 END
```

the following failure results:

```
HCPNFC319E Error processing subsystem command -  
FLASHCOPY for DASD 0101
```

6. The SYNCHRONOUS parameter is not necessary and is ignored when the Enterprise Storage Server FlashCopy Version 2 hardware feature is present on the server device. The ESS hardware FlashCopy



Version 2 feature supports synchronous responses. However, ESS FlashCopy Version 1 does not. When the SYNCHRONOUS parameter is specified and the ESS FlashCopy Version 1 hardware feature is present, CP will simulate a synchronous response.

7. For ESS FlashCopy Version 1 hardware, only one target can be specified. For ESS FlashCopy Version 2 hardware, a maximum of 12 targets can be specified per command. Note that the hardware limits the total number of active FLASHCOPY relationships by both volume and by target. Specifying multiple targets might exceed either of these limits if more than one FLASHCOPY is active on the server.
8. If multiple targets are specified and only FlashCopy Version 1 hardware support is available, you will receive message HCP293E.
9. If you receive an error message from processing, you must verify the state of the target device data.
10. Long-running, multiple target FLASHCOPY commands can be halted by the #CP CPHX command.
11. If both SAVELABEL and LABEL are omitted, the volume label is copied from *vdev1* by the FlashCopy operation if *scy11* and *scy12* are 0.
12. When SAVELABEL or LABEL are specified, the volume label on *vdev2* is always updated, even if *scy12* is nonzero.
13. Failure of LABEL will not prevent the establishment of the relationships.
14. Failure of SAVELABEL will not prevent the establishment of the relationships.
15. Guest FLASHCOPY support is restricted to dedicated or fullpack minidisks.
16. If PRESERVemirror and NOPRESERVemirror are not specified, the Flashcopy operation is attempted first as a PRESERVemirror request. If the first attempt fails, the I/O is retried one time as a NOPRESERVemirror request.
17. FLASHCOPY to Global Mirror (asynchronous PPRC) primary devices is supported.

## Examples

Example 1:

To copy virtual device F101 to F102 which are the same size, enter:

```
FLASHCOPY F101 0 END to F102 0 END
```

## Responses

The following are the command responses issued as a result of successful completion of the FLASHCOPY command:

Response 1: Immediate Response

If the command completes immediately, the following response is displayed:

```
Command complete: FLASHCOPY vdev1 scyl1 ecy11 TO vdev2 scyl2 ecy12
```

There might be one or more completion messages issued when multiple target *vdevs* are issued for the FLASHCOPY command.

- If the *vdev* targets in the command are all contained in one physical volume, only one completion message will be issued using the *vdev* of the last target copied.
- If the *vdev* targets in the command exist on more than one physical volume, the entry order of the targets is important. Each time the next *vdev* in the command list is found to be on a different physical volume from the last target *vdev* that was looked at, an internal FLASHCOPY order and a separate completion message will be issued. Therefore, grouping the *vdev* targets by physical volume will increase command performance and also minimize messages.

Response 2: Delayed Response

If the FLASHCOPY SYNCHRONOUS option is not specified, and only hardware FlashCopy Version 1 support is available, the FLASHCOPY command might not complete immediately. In this case, the following response is displayed:

```
Command started: FLASHCOPY vdev1 scyl1 ecyl1 TO vdev2 scyl2 ecyl2
```

When the copy is complete, the following response is issued:

```
Command complete: FLASHCOPY vdev1 scyl1 ecyl1 TO vdev2 scyl2 ecyl2
```

Under certain circumstances, it is possible that a storage subsystem does not notify CP when it has completed a FLASHCOPY. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing ends after generating the following response:

```
Command results lost: FLASHCOPY vdev1 scyl1 ecyl1 TO vdev2 scyl2 ecyl2
```

## Messages

The messages listed apply to the FLASHCOPY, FLASHCOPY BACKGNDCOPY, FLASHCOPY ESTABLISH, FLASHCOPY RESYNC, FLASHCOPY TGTWRITE, and FLASHCOPY WITHDRAW commands.

Messages:

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP024E Cylinder missing or invalid
- HCP040E Device *vdev* does not exist
- HCP116I WARNING: Cylinders *n* through *n* of minidisk *vdev* are the target of FlashCopy relationship *hhhhhhh*.
- HCP293E Control unit does not support multiple targets
- HCP296E Status is not as required - *vdev*; an unexpected condition occurred while executing a FLASHCOPY command, code = *code*
- HCP297E An unexpected condition occurred after the command was initiated and was accepted by the hardware, code = *code*. *userid* Command: FLASHCOPY *vdev1 scyl1 ecyl1 TO vdev2 scyl2 ecyl2*
- HCP319E Error processing subsystem command - *cmd* for DASD *vdev*
- HCP332E Invalid control unit type - *vdev*
- HCP1600E Specified cylinder number *cyl* is not on the virtual DASD.
- HCP1914E Virtual device *vdev* not write-enabled
- HCP2008E Device *vdev* is currently reserved by user *userid*
- HCP2450E SOURCE operand missing.
- HCP2451E SOURCE must be a full-pack extent.
- HCP2452E SOURCE must be a minidisk extent.
- HCP2453E TARGET operand missing.
- HCP2454E TARGET must be a full-pack extent.
- HCP2455E TARGET must be a minidisk extent.
- HCP2456E SOURCE and TARGET must both be full-pack or minidisk extents.
- HCP2458E SOURCE *vdev* specifies *n* cylinders but TARGET *vdev* specifies *n* cylinders.
- HCP2459E Unable to read or verify the current volume label on *vdev*.
- HCP2460E Unable to update current volume label on *vdev*.
- HCP2461I Volume label on *vdev* is now *volser*.

- HCP2462I *extenttype* extent on CP-OWNED volume *volser*, tracks *cccc/hh* to *cccc/hh*, is the Source for FlashCopy Relationship *sequence*.
- HCP2463E *extenttype* extent on CP-OWNED volume *volser*, tracks *cccc/hh* to *cccc/hh*, is the Target for FlashCopy Relationship *sequence*.
- HCP2464E Volser is missing after LABEL operand.
- HCP2465I No matches found.
- HCP2466I Device *rdev* - No FlashCopy relationships reported.
- HCP2467E The CP-OWNED volume *volser* is allocated on Space-Efficient device *rdev*. Space-Efficient volumes are not allowed as CP-OWNED volumes.
- HCP2468E FlashCopy relationships on CP-OWNED volume *volser* on device *rdev*, cylinders *n* through *n*, have been terminated.
- HCP2469E WITHDRAW failed because *n* tracks remain to be copied to *vdev*.
- HCP2471E At least one of the specified devices does not support PRESERVEMIRROR. Retry the command specifying NOPRESERVEMIRROR.
- HCP3201E DASD Error: Bus-out Parity Check
- HCP3201E DASD Error: Correctable Data Check
- HCP3201E DASD Error: Data Check
- HCP3201E DASD Error: End of Cylinder
- HCP3201E DASD Error: Environmental Data Present
- HCP3201E DASD Error: Equipment Check
- HCP3201E DASD Error: File Protected
- HCP3201E DASD Error: First Error Log
- HCP3201E DASD Error: Imprecise Ending
- HCP3201E DASD Error: Incomplete Domain
- HCP3201E DASD Error: Intervention Required
- HCP3201E DASD Error: Invalid Track Format
- HCP3201E DASD Error: Message to Operator
- HCP3201E DASD Error: No Record Found
- HCP3201E DASD Error: Overrun
- HCP3201E DASD Error: Permanent Error
- HCP3201E DASD Error: Request Inhibit Write
- HCP3201E DASD Error: Sense Information for Remote Copy Services Volume
- HCP3201E DASD Error: Write Inhibited
- HCP3201E DASD Error: Command Reject  
Message 0: No specific message code
- HCP3201E DASD Error: Command Reject  
Message 1: Invalid command
- HCP3201E DASD Error: Command Reject  
Message 2: Invalid command sequence
- HCP3201E DASD Error: Command Reject  
Message 3: CCW count less than required
- HCP3201E DASD Error: Command Reject  
Message 4: Invalid parameter (Reason Code=*pp qq qq rr ss*)
- HCP3201E DASD Error: Command Reject  
Message 5: Diagnostic or special command violates file mask
- HCP3201E DASD Error: Command Reject

- Message 6: Unknown Reason Code from Hardware
- HCP3201E DASD Error: Command Reject  
Message 7: Channel returned with incorrect retry CCW
  - HCP3201E DASD Error: Command Reject  
Message 8: Reset notification
  - HCP3201E DASD Error: Command Reject  
Message 9: Storage Control restart
  - HCP3201E DASD Error: Command Reject  
Message A: Channel problem
  - HCP3201E DASD Error: Command Reject  
Message B: Invalid defective/alternate track pointer
  - HCP3201E DASD Error: Command Reject  
Message C: Unexpected message code from hardware
  - HCP3201E DASD Error: Command Reject  
Message D: Summary unit check
  - HCP3201E DASD Error: Command Reject  
Message E: Command invalid on PPRC secondary address
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 05: The addressed volume size is greater than the maximum allowed for the operation.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 0D: Operation requires nonvolatile storage and nonvolatile storage is not available.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 22: A specified device is unconfigured, in CE Mode, or status cannot be determined.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 37: A command was received while the subsystem was attempting to install a new microcode load.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 56: The source or the target volume has pinned data.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 57: A specified device is in a suspended Peer-to-Peer Remote Copy state. All write commands with normal authorization indicated are rejected.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 58: An operation was received that conflicts with an established Peer-to-Peer Remote Copy state for the device.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 60: The target volume for a FlashCopy operation has grouped paths (that is, online to host) which conflicts with the setting that disallows the operation if the volume is online.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 65: A FlashCopy or Peer-to-Peer Remote Copy operation could not complete because the target is reserved to another initiator or the target is not ready.
  - HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 66: A volume with the requested track format could not be allowed.

- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 6B: The operation cannot be executed when the channel is operating in Transport Mode and the Service Action Code is X'1FFE'.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 6C: The operation cannot be executed when the channel is operating in Transport Mode and the Service Action Code is X'1FFF'.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 82: No resources available to create relationships for the FlashCopy operation or the number of relationships will exceed the maximum allowed.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 83: The battery feature is unavailable or not installed. FlashCopy requires the battery for proper operation.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 84: The FlashCopy volumes specified are not the same track type, the target volume has fewer tracks than the source, the number of target extents does not equal the number of source, or the source and target are not in the same logical subsystem (on some early FlashCopy Version 1 hardware).
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 85: The target FlashCopy volume is a Peer-to-Peer Remote Copy primary or secondary or is an XRC primary volume.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 86: The feature that the operation references is not installed or enabled for the subsystem.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 89: Establish FlashCopy specified a preserve mirror operation. A condition prevents the preserve mirror operations from being successful. Reason Code Qualifier: qr.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 8A: A Resync FlashCopy operation was received and the volumes were not in a persistent relationship whose background copy had completed.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 8E: A FlashCopy operation specified a target volume that is an Asynchronous Peer-to-Peer Remote Copy primary volume.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 98: The target volume of a FlashCopy operation is a Peer-to-Peer Remote Copy primary volume and is not in the correct Peer-to-Peer Remote Copy state.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 99: An operation was received that is not supported by the hardware or software level of the DASD subsystem.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 9B: A FlashCopy or Peer-to-Peer Remote Copy Pair operation was rejected because the target is reserved to another initiator.
- HCP3201E DASD Error: Command Reject

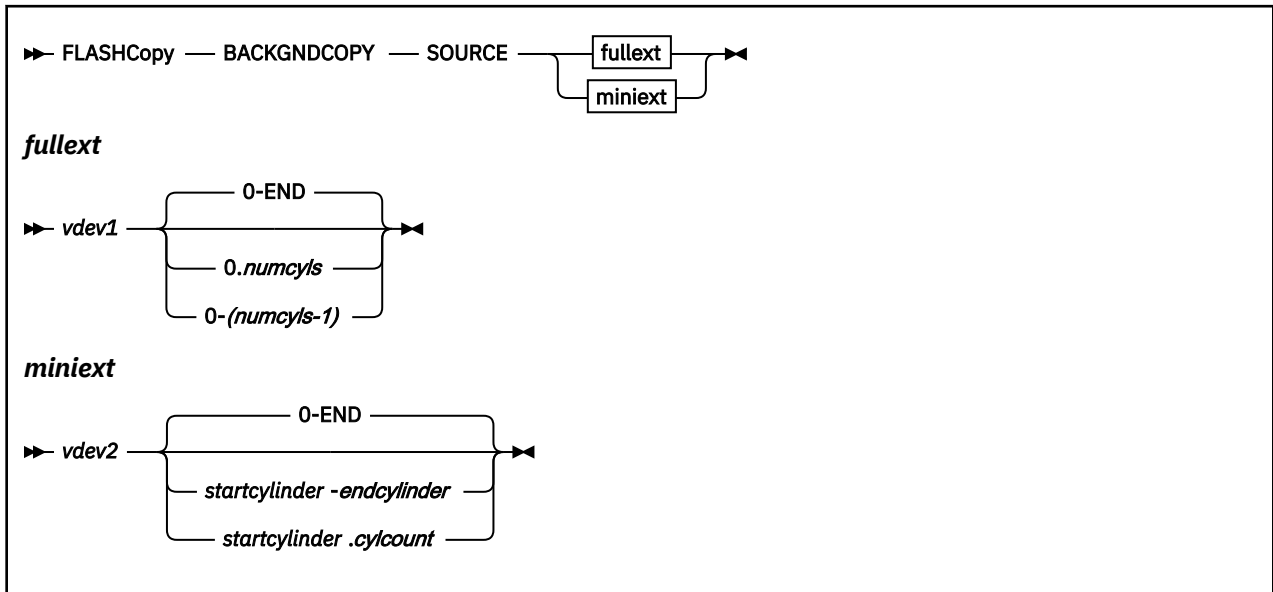
Message F: Status not as required

Reason 9C: A FlashCopy operation cannot complete due to internal hardware conditions or lack of resources to manage the relationships.

- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason 9D: A FlashCopy operation cannot complete because the maximum number of relationships for the volume has been exceeded; FlashCopy Version 2 (Data Set level) only.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason A5: The FlashCopy operation does not allow the specification of extents; FlashCopy Version 1 (Full Volume) only.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason A6: A FlashCopy operation cannot complete because the maximum number of FlashCopy target relationships has been exceeded for one or more of the specified source tracks.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason A7: A FlashCopy operation cannot complete because it has specified that an existing target track become a source track. Any single track must not be both a source and a target and a target track can only have one source.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason A8: A FlashCopy operation cannot complete because the maximum number of relationships for the volume has been exceeded.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason A9: A FlashCopy operation was received while a volume was inhibited for FlashCopy relationships.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason AB: A FlashCopy Version 1 (Full Volume) operation was received on a volume that has FlashCopy Version 2 (Data Set Level) relationships.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason AC: A FlashCopy operation was received that is no longer supported.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason AD: A FlashCopy operation cannot be completed because the specified volumes has no grouped paths (that is, offline to host) or is in an inaccessible state.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason AE: A FlashCopy operation cannot complete because it has specified that an existing target track become a target track. Any single track must not be both a source and a target and a target track can only have one source.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason AF: A FlashCopy operation cannot complete because it has specified that an existing source track become a target track. Any single track must not be both a source and a target and a target track can only have one source.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason B4: A FlashCopy operation attempted to reverse the order of the FlashCopy relationship and the target bitmap was not all zeros (that is, changes pending).

- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason B6: The state of an existing FlashCopy relationship does not allow the specified FlashCopy operation.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason B7: A FlashCopy operation was received and VM minidisk mode was specified, but an existing FlashCopy relationship does not have VM minidisk mode specified, or vice versa.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason B9: A FlashCopy operation attempted to split an existing full volume incremental relationship.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason BA: Change Recording has been specified by a FlashCopy operation, but Change Recording is already active and was not previously activated by the same operation type.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason BB: A FlashCopy operation indicated Terminate All Relationships but the specified extents contain an existing full volume source relationship.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason BC: A FlashCopy operation specifying Fast Reverse Restore was received while a previous Fast Reverse Restore operation was in progress.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason BD: A FlashCopy operation specified an option that is not valid with a space efficient volume.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason BE: A FlashCopy operation failed because the specified repository was offline.
- HCP3201E DASD Error: Command Reject  
Message F: Status not as required  
Reason EF: Reserved for software - Subchannel ID mismatch.
- HCP6706E Invalid cylinder number - *cyl*
- HCP6706E Invalid device address - *vdev*
- HCP6790I FLASHCOPY processing halted. The next device that would have been processed is *vdev*

## FLASHCOPY BACKGNDCOPY



### Authorization

Privilege Class: B

### Purpose

For non-space-efficient relationships, use FLASHCOPY BACKGNDCOPY to initiate background copying of unmodified source data to its targets. This command reverses the effect of the NOCOPY option of the FLASHCOPY ESTABLISH command. Refer to [Track Space-Efficient Volumes and FlashCopy Backups in z/VM: CP Planning and Administration](#) for more information regarding background copying.

### Operands

#### SOURCE

specifies the readable disk and range of cylinders on a dedicated device, full-pack minidisk, or non-full-pack minidisk.

#### vdev1

defines a dedicated device or full-pack minidisk.

#### 0-END

identifies all the real cylinders of the dedicated device or full-pack minidisk.

#### numcyls

identifies the number of real cylinders in this extent. The value must be equal to the number of cylinders defined for the dedicated device or full-pack minidisk.

#### vdev2

defines a non-full-pack minidisk.

#### 0-END

identifies all the virtual cylinders of the minidisk.

#### startcylinder

identifies the first virtual cylinder number for this extent, starting with 0 to indicate the first cylinder. This can be a value from 0 to the number of cylinders on the minidisk minus 1.



**endcylinder**

identifies the last virtual cylinder number for this extent, starting with 0 to indicate the first cylinder. This can be a value from 0 to the number of cylinders on the minidisk minus 1 and must be greater than or equal to *startcylinder*.

**cylcount**

identifies the number of cylinders in this extent. The value of *startcylinder* plus *cylcount* must be less than or equal to the number of cylinders defined for the minidisk.

**Usage Notes**

1. This command has no effect if the relationship was not established with the implied or explicit NOCOPY option.
2. Issuing the command to any source cylinder within a relationship will cause all of the referenced relationship to begin its background copy operations. For example, if you specify:

```
FLASHCOPY ESTABLISH SOURCE 200 TARGET 300 310 NOCOPY
FLASHCOPY ESTABLISH SOURCE 200 TARGET 320 NOCOPY
FLASHCOPY BACKGNDCOPY SOURCE 200 1.1
```

the background copying will begin for all cylinders for targets 300, 310, and 320.

3. Multiple executions of this command to the same source extent have no additional effect.
4. Copying of the data from the source to its targets might not occur immediately, depending on the availability of storage subsystem resources.
5. It is suggested, but not required, that the NOSETARGET operand be specified when establishing a relationship for use with this command.
6. If one or more affected targets are track space-efficient volumes, the command will be rejected by the hardware. However, this will NOT prevent the background copy from being started on any targets that are NOT track space-efficient.

**Examples**

## Example 1:

```
define mdisk 0200 100 100 pack01
define mdisk 0300 100 100 pack02

flashcopy establish source 200 target 300 nocopy
Command complete: FLASHCOPY 0200 0 99 TO 0300 0 99
FLASHCOPY RELATIONSHIP 4A7AC71E ESTABLISHED
Ready; T=0.01/0.01 12:28:08

flashcopy withdraw target 300
HCPNFC2469E WITHDRAW failed because 1500 tracks remain to be copied to 300.
Ready(02469); T=0.01/0.01 12:28:10

flashcopy backgndcopy source 200
BACKGROUND COPY STARTED
Ready; T=0.01/0.01 12:28:19

flashcopy withdraw target 300
RELATIONSHIP WITHDRAW
Ready; T=0.01/0.01 12:29:00
```

**Responses**

## Response 1:

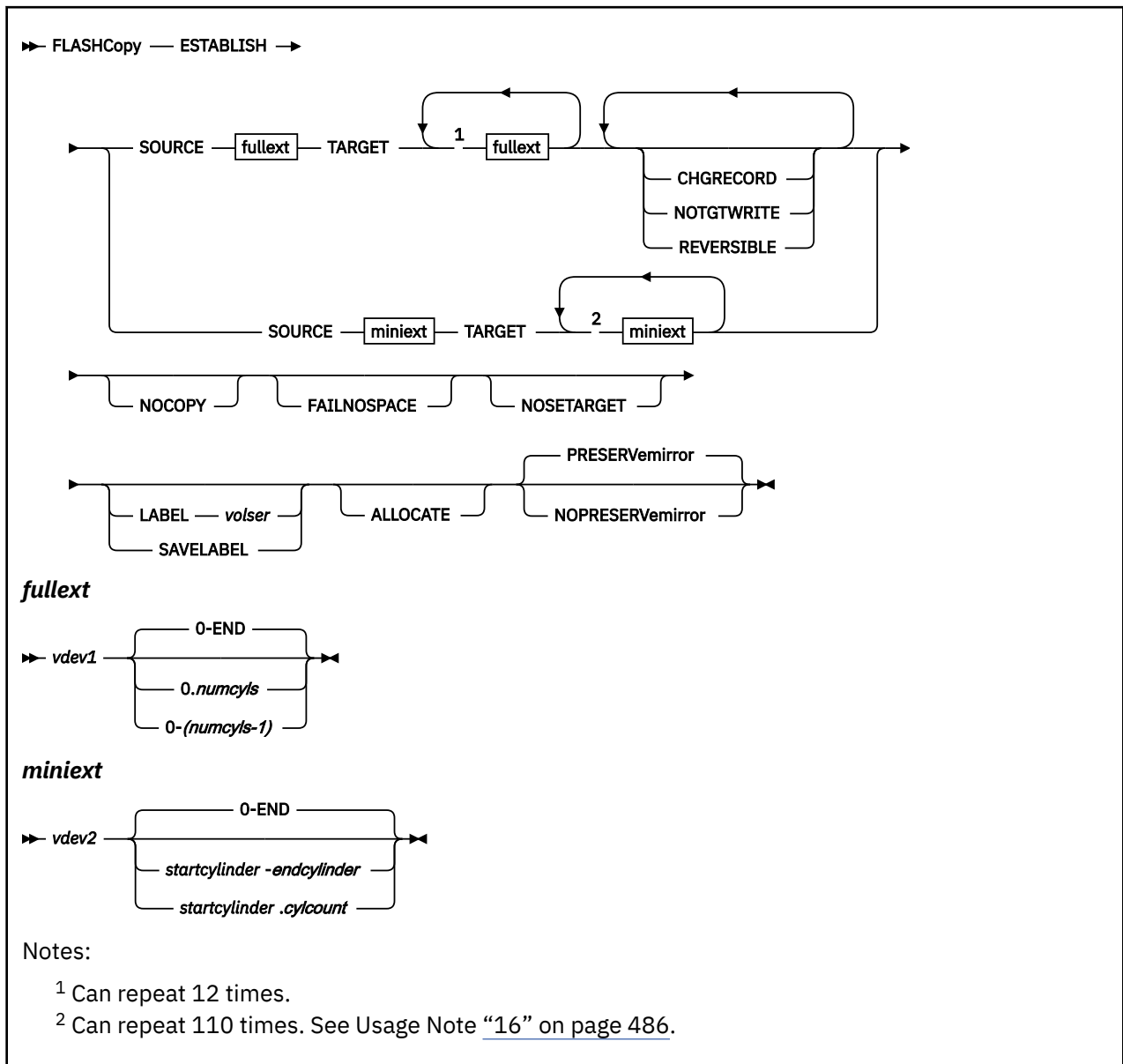
This FLASHCOPY BACKGNDCOPY command completed successfully:

```
flashcopy backgndcopy source 200
BACKGROUND COPY STARTED
Ready; T=0.01/0.01 12:28:19
```

### Messages

For a complete list of messages that can be issued by the FLASHCOPY BACKGNDCOPY command, see the [“FLASHCOPY” on page 471](#) command.

## FLASHCOPY ESTABLISH



### Authorization

Privilege Class: B

### Purpose

Use FLASHCOPY ESTABLISH to create a persistent FlashCopy relationship for point-in-time backups and test data replication scenarios.

### Operands

#### SOURCE

specifies the readable disk and range of cylinders that are to be copied by the FLASHCOPY command.

#### vdev1

defines a dedicated device or full-pack minidisk.

**O-END**

identifies all the real cylinders of the dedicated device or full-pack minidisk.

***numcyls***

identifies the number of real cylinders in this extent. The value must be equal to the number of cylinders that are defined for the dedicated device or full-pack minidisk.

**TARGET**

specifies the writable disks and extents that are to be copied by the FLASHCOPY command. You can supply up to 12 unique target *vdevs* and up to a total of 110 extents.

***vdev2***

defines a non-full-pack minidisk.

**O-END**

identifies all the virtual cylinders of the minidisk.

***startcylinder***

identifies the first virtual cylinder number for this extent, starting with 0 to indicate the first cylinder. This cylinder number can be a value from 0 to the number of cylinders on the minidisk minus 1.

***endcylinder***

identifies the last virtual cylinder number for this extent, starting with 0 to indicate the first cylinder. This cylinder number can be a value from 0 to the number of cylinders on the minidisk minus 1 and must be greater than or equal to *startcylinder*.

***cylcount***

identifies the number of cylinders in this extent. The value of *startcylinder* plus *cylcount* must be less than or equal to the number of cylinders defined for the minidisk.

**CHGRECORD**

creates an initial checkpoint for a subsequent FLASHCOPY RESYNC command. This option is implied by specifying the REVERSIBLE option.

**NOTGTWRITE**

inhibits writes on the target volume. This setting can be later changed with a FLASHCOPY TGTWRITE command. This option is implied when REVERSIBLE is specified. This option is mutually exclusive with the LABEL or SAVELABEL options.

**REVERSIBLE**

builds a reversible FLASHCOPY relationship that can later be altered by a FLASHCOPY RESYNC command with a reversed source and target. This option is mutually exclusive with the NOCOPY option and implies that CHGRECORD and NOTGTWRITE were specified. This option is mutually exclusive with the LABEL and SAVELABEL options.

Space-efficient volumes should not be used in reversible relationships.

**NOCOPY**

prevents the immediate background copying of unmodified SOURCE tracks to the targets. See [Using Persistent FlashCopy in z/VM: CP Planning and Administration](#) for a more detailed discussion of this feature. Note that the data might be copied despite the specification of the NOCOPY option. This option is implied when the target is a space-efficient volume. The NOCOPY option is mutually exclusive with the REVERSIBLE option.

**FAILNOSPACE**

causes the relationship to fail if the space-efficient target becomes out of space. This option is the default when the target is a space-efficient volume.

**NOSETARGET**

verifies that all target extents are NOT on space-efficient volumes. This option should be specified if you anticipate running a subsequent FLASHCOPY BACKGNDCOPY command. This option should also be used by applications that do not understand how to handle space-efficient out-of-space conditions. The default action is to allow space-efficient targets. If the specified target is a space-efficient volume, Reason Code BD is returned. This option is implied by specifying the REVERSIBLE option.

**LABEL *volser***

asks CP to change the volume label on the virtual cylinder 0, head 0, record 3 of targets to *volser*. This option reads and verifies the volume label on the target virtual cylinder 0, head 0, record 3 **after** the relationship is established. This implies that the target must contain a valid label that was either copied from the source or not overwritten at all. After the volume label is verified, it is rewritten with the specified *volser* onto target cylinder 0, head 0, record 3.

This option is mutually exclusive with the NOTGTWRITE and REVERSIBLE options.

**SAVELABEL**

tells CP to preserve the volume label on target virtual cylinder 0, head 0, record 3. This option attempts to read and verify the volume label from the target devices **before** the relationship is established. This means that a volume label should exist on the target before this command is issued. Note that initially space-efficient targets typically do not have any cylinders and therefore do not have a label record that can be read. Therefore, SAVELABEL with space-efficient targets typically fails with message HCP2459E. After the relationship is established, the same processing as for the LABEL option is completed to update the label.

This option is mutually exclusive with the NOTGTWRITE and REVERSIBLE options.

**ALLOCATE**

requests that the physical space required on a target Extent Space Efficient volume is allocated before the completion of the ESTABLISH command.

**PRESERVEmirror**

requests that the FlashCopy operation must not change the target PPRC pair to a duplex-pending status.

**NOPRESERVEmirror**

requests that the FlashCopy operation can complete with no requirements on the PPRC status of the target devices.

**Usage Notes**

1. A TARGET up to 12 *vdevs* can be specified with a limit of 110 extent descriptions that are shared among them.
2. Some operands are restricted to dedicated devices or full-pack minidisks. Restrictions can change as new hardware releases occur.
3. You cannot mix *fullext* and *miniext* specifications in a command, even if they describe areas of the same total space.
4. Each source cylinder can be the source for up to 12 relationships. This limit is enforced even when multiple FLASHCOPY ESTABLISH commands are issued. For example, if you specify:

```
FLASHCOPY ESTABLISH SOURCE 0200 TARGET 0300
```

and

```
FLASHCOPY ESTABLISH SOURCE 0200 TARGET 0310
```

you now have two of the limit of 12 relationships on 0200. This limit also applies to relationships established on other LPARS.

5. Specifying:

```
FLASHCOPY ESTABLISH SOURCE 0200 TARGET 0300 0310
```

is not quite the same as:

```
FLASHCOPY ESTABLISH SOURCE 0200 TARGET 0300
FLASHCOPY ESTABLISH SOURCE 0200 TARGET 0310
```

The difference is that in the first case, one sequence number that is assigned to the relationship encompasses all targets. In the second case, separate relationships have unique sequence numbers.

6. Cylinders are copied in the order they are specified. For example, if you specify:

```
FLASHCOPY ESTABLISH SOURCE 200 0.100TARGET 300 50.50 300 0.50
```

cylinders 0 - 49 are copied to cylinders 50 - 99, and cylinders 50 - 99 are copied to cylinders 0 - 49.

7. A T-disk (DEFINE T3390...) cannot be used as a persistent FlashCopy source or target.
8. Persistent FlashCopy relationships are visible to all LPARs that are attached to the Storage LSS.
9. It is possible that a persistent relationship that is created on one LPAR can be withdrawn from another.
10. If both SAVELABEL and LABEL are omitted, the volume label is copied from *vdev1* by the FlashCopy operation if source virtual cylinder 0 is copied onto target virtual cylinder 0. The volume label is copied to the corresponding target cylinder.
11. When SAVELABEL or LABEL is specified, the volume label is always updated on target virtual cylinder 0, even if the target virtual cylinder 0 was not copied.
12. Failure of LABEL will not prevent the establishment of the relationships.
13. Failure of SAVELABEL does not prevent the establishment of relationships.
14. FLASHCOPY ESTABLISH is not supported on PAV or HYPERPAV ALIAS *vdevs*.
15. FlashCopy to track space-efficient targets is only supported by hardware with full volume targets.
16. Due to the CP command-line length limit of 240 characters, it might not be possible to specify the full 110 extents in some command entry modes.
17. When a relationship is created with the NOTGTWRITE option, any attempt to write on the volume (other than by FlashCopy itself) places the volume into the Intervention Required state. The situation can cause the writing virtual machine to stop.
18. If PRESERVemirror and NOPRESERVemirror are not specified, the FlashCopy operation is attempted first as a PRESERVemirror request. If the first attempt fails, the I/O is retried one time as a NOPRESERVemirror request.
19. FLASHCOPY to Global Mirror (asynchronous PPRC) primary devices is supported.

## Examples

Example 1: Basic command

```
flashcopy establish source 200 target 300
Command complete: FLASHCOPY 0200 0 99 TO 0300 0 99
FLASHCOPY RELATIONSHIP 4A07BA0F ESTABLISHED
Ready; T=0.01/0.01 10:19:27

q flashcopy hardware 5100
-----SOURCE----- TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
4A07BA0F 1800 5100 PACK01 100/00 5101 PACK02 100/00 0/1500
Ready; T=0.01/0.01 10:19:30
```

**For More Information**, see [Using Persistent FlashCopy in z/VM: CP Planning and Administration](#) for a detailed discussion of potential uses of persistent FlashCopy relationships.

Example 2: Multiple Copies

When you specify multiple target *vdevs*, the total number of cylinders that are specified or implied for each target *vdev* must be equal to the number of cylinders that are specified or implied for the source. For example, if you specify: SOURCE 200 0.50 TARGET 300 0.25 310 0.25, you receive an error:

```
flashcopy establish source 200 0.50 target 300 0.25 310 0.25
HCPNFC2458E SOURCE 0200 specifies 50 cylinders but TARGET 0300 specifies 25
cylinders.
Ready(02458); T=0.01/0.01 15:49:49
```

However, SOURCE 200 0.50 TARGET 300 0.20 310 0.50 300 20.30 is valid:

```
flashcopy establish source 200 0.50 target 300 0.20 310 0.50 300 20.30
Command complete: FLASHCOPY 0200 0 19 TO 0300 0 19
Command complete: FLASHCOPY 0200 20 49 TO 0300 20 49
Command complete: FLASHCOPY 0200 0 49 TO 0310 0 49
FLASHCOPY RELATIONSHIP E796AB5B ESTABLISHED
Ready; T=0.01/0.01 15:54:06
```

You can mix and match the extents, but in the end the total number of cylinders for each *vdev* must be the same. To accomplish what was intended in the first example, you can use the following parameters:

```
SOURCE 200 0.25 TARGET 300 0.25
```

and then the following parameters:

```
SOURCE 200 25.25 TARGET 310 0.25
```

However, these actions yield two relationships, not one.

### Example 3: Cylinder Rearranging

```
flashcopy establish source 200 0.100 target 300 50.50 300 0.50
Command complete: FLASHCOPY 0200 0 49 TO 0300 50 99
Command complete: FLASHCOPY 0200 50 99 TO 0300 0 49
FLASHCOPY RELATIONSHIP 4A04A166 ESTABLISHED
Ready; T=0.01/0.01 10:16:00
```

### Example 4: Multiple Copies With Cylinder Rearranging

```
flashcopy establish source 200 target 300 50.50 310 75.25 300 0.50 310 0.75
Command complete: FLASHCOPY 0200 0 49 TO 0300 50 99
Command complete: FLASHCOPY 0200 50 99 TO 0300 0 49
Command complete: FLASHCOPY 0200 0 24 TO 0310 75 99
Command complete: FLASHCOPY 0200 25 99 TO 0310 0 74
FLASHCOPY RELATIONSHIP 4A540B4F ESTABLISHED
Ready; T=0.01/0.01 11:44:49
```

```
q flashcopy hardware 5101
-----SOURCE----- TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
4A540B4F 8800 5100 PACK01 150/00 5101 PACK02 100/00 0/750
4A540B4F 8800 5100 PACK01 100/00 5101 PACK02 150/00 0/750
4A540B4F 8800 5100 PACK01 125/00 5101 PACK02 200/00 0/1125
4A540B4F 8800 5100 PACK01 100/00 5101 PACK02 275/00 0/375
Ready; T=0.01/0.01 10:45:47
```

## Responses

Response 1:

This FLASHCOPY ESTABLISH command completed successfully:

```
flashcopy establish source 200 12-24 target 300 0-4 300 20.7 300 7.1
Command complete: FLASHCOPY 0200 12 16 TO 0200 0 4
Command complete: FLASHCOPY 0200 17 23 TO 0200 20 26
Command complete: FLASHCOPY 0200 24 24 TO 0200 7 7
FLASHCOPY RELATIONSHIP 58BA8440 ESTABLISHED
Ready; T=0.01/0.01 11:15:03
```

Response 2:

It is possible for a FLASHCOPY ESTABLISH command to partially complete and then fail. In such cases, when any of the requested relationships are formed, the relationship ID is presented along with a nonzero return code.

To verify that no relationships exist on the target volume:

```
q flashcopy hardware 5101
HCPQFL2466E Device 5101 - No FlashCopy relationships reported.
Ready(02466); T=0.01/0.01 16:01:20
```

## FLASHCOPY ESTABLISH

To issue an ordinary command:

```
flashcopy establish source 200 target 300
Command complete: FLASHCOPY 0200 0 99 TO 0300 0 99
FLASHCOPY RELATIONSHIP E79D8AB4 ESTABLISHED
Ready; T=0.01/0.01 16:01:47
```

To issue a command that cannot succeed because you have duplicate targets:

```
flashcopy establish source 200 target 310 300
Command complete: FLASHCOPY 0200 0 99 TO 310 0 99
HCPPFC3201E: DASD Error: Command Reject
HCPPFC3201E: Message F: Status not as required
HCPPFC3201E: Reason AE: A FlashCopy operation cannot complete because
it has specified that an existing target track become a target track.
Any single track must not be both a source and a target and a target
track can only have one source.
FLASHCOPY RELATIONSHIP E79DA36F ESTABLISHED
Ready(03201); T=0.01/0.01 16:01:55
```

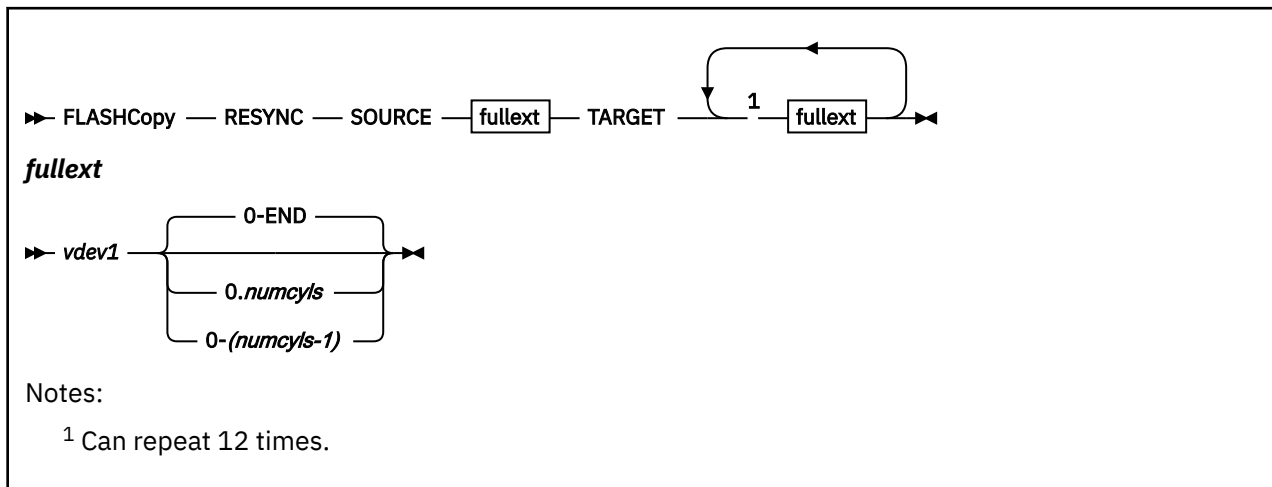
```
q flashcopy hardware 5101
-----SOURCE----- -----TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
E79D8AB4 8800 5100 RWS001 100/00 5101 RWS002 100/00 0/1500
E79DA36F 8800 5100 RWS001 100/00 5101 RWS002 200/00 0/1500
Ready; T=0.01/0.01 16:02:00
```

## Messages

For a list of general messages that can be issued by the FLASHCOPY ESTABLISH command, see the [“FLASHCOPY” on page 471](#) command.



## FLASHCOPY RESYNC



### Authorization

Privilege Class: B

### Purpose

Use FLASHCOPY RESYNC to resynchronize the specified target of a persistent checkpointed FlashCopy relationship to make the specified target consistent with the current state of the specified source and to establish a new checkpoint. The source volume is not changed.

Checkpoints are established at the completion of the FLASHCOPY ESTABLISH CHGRECORD, FLASHCOPY ESTABLISH REVERSIBLE, AND FLASHCOPY RESYNC commands.

Table 19. FlashCopy Resync Operation

Source Track Changed	Target Track Changed	Action
N	N	None
N	Y	Target track rolled back to current source track
Y	N	Target track rolled forward from current source track
Y	Y	Target track rolled back from current source track

Reversible RESYNC:

A relationship can be reversed by specifying the corresponding source and target operands reversed from their specification during a prior ESTABLISH or RESYNC like this:

```
flashcopy establish source fce0 target fee0 reversible
Command complete: FLASHCOPY FCE0 0 END TO FEE0 0 END
FLASHCOPY RELATIONSHIP 2F41267E ESTABLISHED
Ready; T=0.01/0.01 16:15:51

q flashcopy hardware fce0
-----SOURCE-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
2F41267E 5900 FCE0 RWSCE0 0/00 FEE0 RWSEE0 0/00 0/50085
Ready; T=0.01/0.01 16:16:08

flashcopy resync source fee0 target fce0
RESYNC COMPLETED
Ready; T=0.01/0.01 16:16:30
```

```

q flashcopy hardware fce0
-----SOURCE----- TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
2F41BC80 C900 FEE0 RWSCE0      0/00 FCE0 RWSCE0      0/00 0/50085
Ready; T=0.01/0.01 16:16:32

```

In the FLGS of the QUERY FLASHCOPY HARDWARE command in the above example, the source of the relationship has become FEE0 and the target has become FCE0.

Prior to reversing a relationship, all of the tracks must have been copied (using the Background Copy Function) onto the target. If this condition is not met, you will receive error B4. In the following example, the FLASHCOPY ESTABLISH command specified NOCOPY which prevented the tracks from being copied to the original target.

```

FLASHCOPY ESTABLISH SOURCE FCE0 TARGET FEE0 CHGRECORD NOCOPY
Command complete: FLASHCOPY FCE0 0 END TO FEE0 0 END
FLASHCOPY RELATIONSHIP 2F267DD5 ESTABLISHED
Ready; T=0.01/0.01 16:15:51

q flashcopy hardware fce0
-----SOURCE----- TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
2F267DD5 6918 FCE0 RWSCE0      0/00 FEE0 RWSCE0      0/00 49795/50085
Ready; T=0.01/0.01 15:46:45

```

Here is the resulting error:

```

FLASHCOPY RESYNC SOURCE FEE0 TARGET FCE0
HCPPFC3201E DASD Error: Command Reject
HCPPFC3201E Message F: Status not as required
HCPPFC3201E Reason B4: A FlashCopy operation attempted to reverse the
order of the FlashCopy relationship and the target bitmap was not all
zeros (that is, changes pending).
Ready(03201); T=0.01/0.01 15:46:37

```

Because of this limitation, the use of the NOCOPY option is discouraged when you are planning to use reversible relationships. Also, because NOCOPY is forced for space-efficient volumes, then space-efficient volumes cannot be used in this context.

To assist with correctly creating relationships that are reversible, the REVERSIBLE option of the FLASHCOPY ESTABLISH command should be specified.

## Operands

### SOURCE

describes a FlashCopy source volume that is a dedicated device or full-pack minidisk. This is the master volume for the resynchronization.

### TARGET

describes the target volumes that are dedicated devices or full-pack minidisks that are to be resynchronized to the source master volume.

### *vdev1*

defines a dedicated device or full-pack minidisk.

### **0-END**

identifies all the real cylinders of the dedicated device or full-pack minidisk.

### *numcyls*

identifies the number of real cylinders in this extent. The value must be equal to the number of cylinders defined for the dedicated device or full-pack minidisk.

## Usage Notes

1. This relationship is not withdrawn.
2. It is possible to reverse a relationship by specifying the current source and target volumes reversed from their original specification. To accomplish this, the relationship must have been ESTABLISHED with the REVERSIBLE option.

3. The relationship to be resynchronized must have been created with the CHGRECORD or REVERSIBLE option.
4. It does not matter in what order the changes are made to the tracks. For example, if you change source track 100 to value 'X' and later change target track 100 to value 'Y', the RESYNC operation will end with target track 100 having a value of 'X'. In other words, the RESYNC operation proceeds based solely upon the current state of the volumes, not the order in which changes were made.
5. Space-efficient volumes cannot be used for reversible RESYNC operations.
6. We recommend that you do not update the target volume while a RESYNC operation is in progress because some tracks might be written by the application and then rewritten by RESYNC processing. This will likely result in creating a volume that is a mixture of the source and the updates that are being attempted.

## Examples

Example 1:

```
flashcopy establish source 0200 target 0300
flashcopy resync source 0200 target 0300

flashcopy establish source 0200 target 0300 reversible
flashcopy resync source 0300 target 0200
```

## Responses

Response 1:

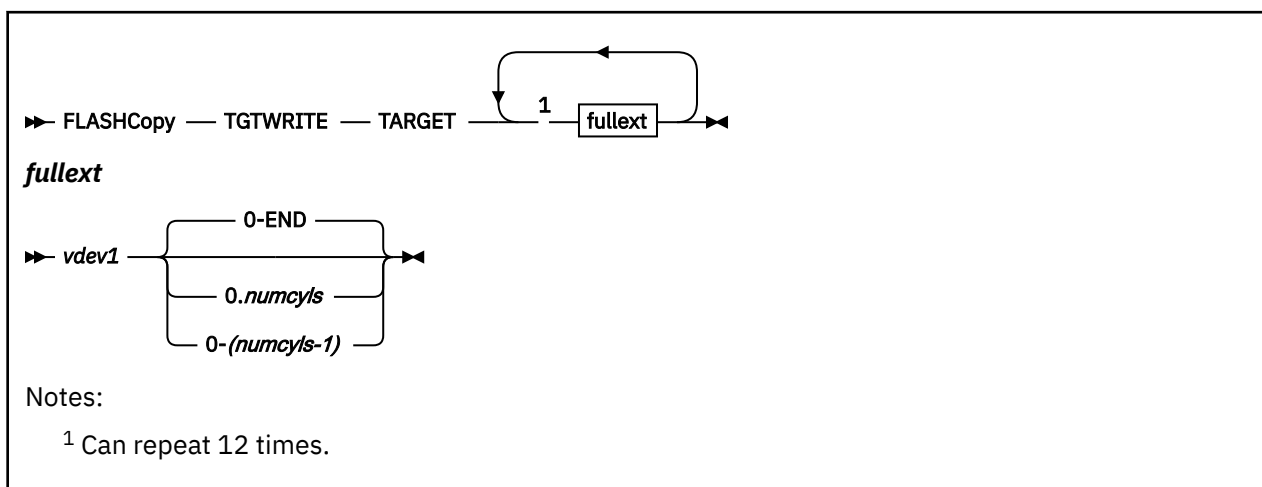
This FLASHCOPY RESYNC command completed successfully:

```
flashcopy resync source 0200 target 0300
RESYNC COMPLETED
Ready;
```

## Messages

For a complete list of messages that can be issued by the FLASHCOPY RESYNC command, see the [“FLASHCOPY”](#) on page 471 command.

## FLASHCOPY TGTWRITE



### Authorization

Privilege Class: B

### Purpose

Use FLASHCOPY TGTWRITE to enable writing on a target full volume that was write-inhibited with the FLASHCOPY ESTABLISH NOTGTWRITE option.

### Operands

#### TARGET

describes the volumes that are dedicated devices or full-pack minidisks that are to be affected.

#### vdev1

defines a dedicated device or full-pack minidisk.

#### 0-END

identifies all the real cylinders of the dedicated device or full-pack minidisk.

#### numcyls

identifies the number of real cylinders in this extent. The value must be equal to the number of cylinders defined for the dedicated device or full-pack minidisk.

### Usage Notes

1. This relationship will not be withdrawn.
2. Applying this command to a REVERSIBLE relationship makes it no longer reversible.

### Examples

Example 1:

```
flashcopy tgtwrite 0300
```

### Responses

Response 1:

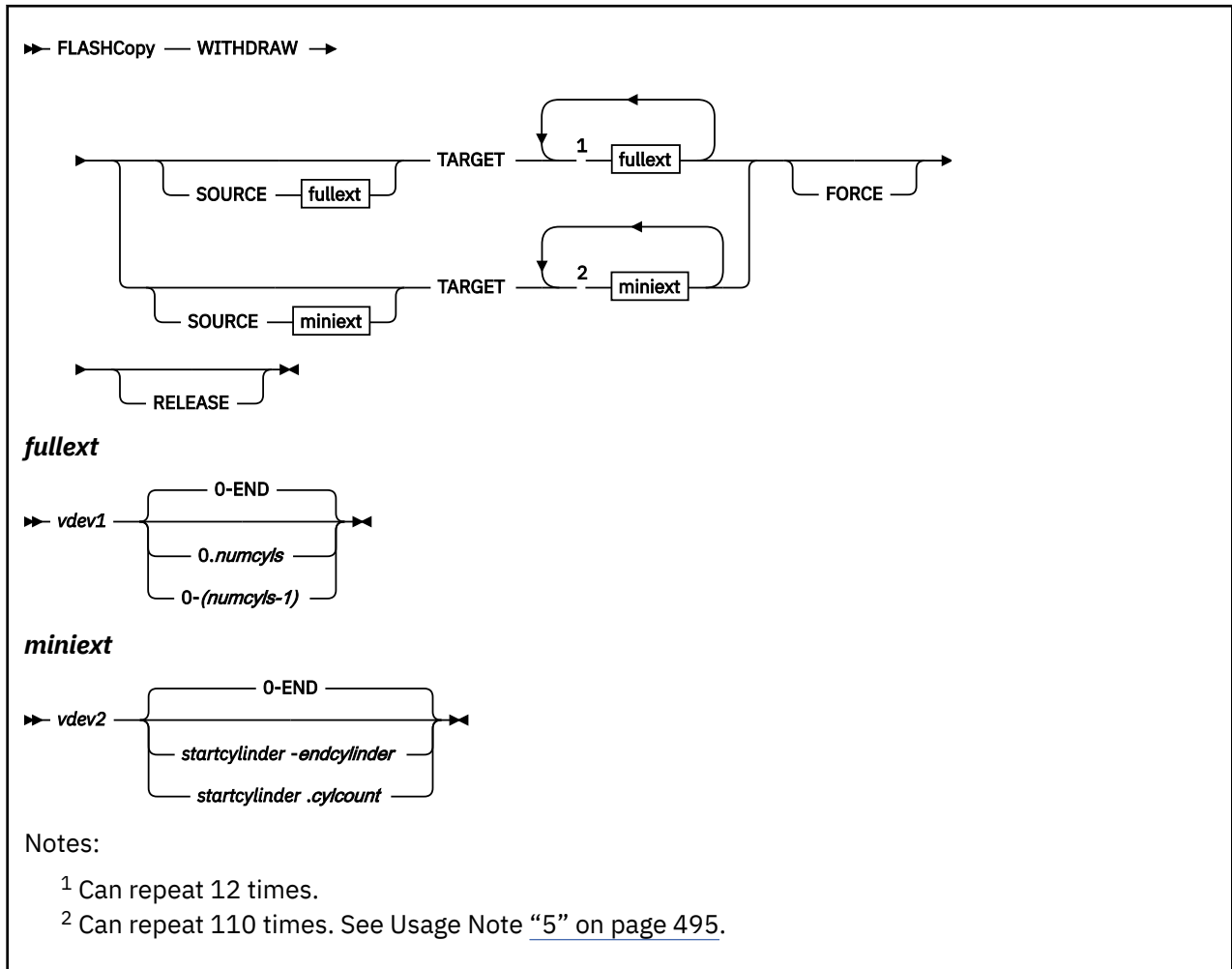
This FLASHCOPY TGTWRITE command completed successfully:

```
flashcopy tgtwrite 0300  
TARGET WRITING NOW ENABLED  
Ready;
```

## Messages

For a complete list of messages that can be issued by the FLASHCOPY TGTWRITE command, see the [“FLASHCOPY” on page 471](#) command.

## FLASHCOPY WITHDRAW



### Authorization

Privilege Class: B

### Purpose

Use FLASHCOPY WITHDRAW to remove a persistent FlashCopy relationship.

### Operands

#### SOURCE

specifies the readable disk and range of cylinders on a dedicated device, full-pack minidisk, or non-full-pack minidisk. If specified, those tracks that are within the source extent and target extent are withdrawn. If source is not specified, the appropriate source extents are determined from the targets.

#### vdev1

defines a dedicated device or full-pack minidisk.

#### 0-END

identifies all the real cylinders of the dedicated device or full-pack minidisk.

**numcyls**

identifies the number of real cylinders in this extent. The value must be equal to the number of cylinders defined for the dedicated device or full-pack minidisk.

**TARGET**

specifies the writable disks and extents that are to be withdrawn. Not all of the cylinders within the specified extents are required to be in an active persistent relationship.

**vdev2**

defines a non-full-pack minidisk.

**0-END**

identifies all the virtual cylinders of the minidisk.

**startcylinder**

identifies the first virtual cylinder number for this extent, starting with 0 to indicate the first cylinder. This can be a value from 0 to the number of cylinders on the minidisk minus 1.

**endcylinder**

identifies the last virtual cylinder number for this extent, starting with 0 to indicate the first cylinder. This can be a value from 0 to the number of cylinders on the minidisk minus 1 and must be greater than or equal to *startcylinder*.

**cylcount**

identifies the number of cylinders in this extent. The value of *startcylinder* plus *cylcount* must be less than or equal to the number of cylinders defined for the minidisk.

**FORCE**

specifies the immediate and unconditional withdrawal of the specified relationships. When FORCE is specified, the target extents might not be usable because the SOURCE data might not have been completely copied.

**RELEASE**

specifies that all TARGET space-efficient tracks or extents are to be destroyed and returned to the space-efficient repository for reassignment.

**Usage Notes**

1. When the last target extent is withdrawn, the corresponding source relationships are withdrawn automatically.
2. If background source-to-target copies are pending and the relationship was established with NOCOPY, you must issue a separate FLASHCOPY BACKGNDCOPY command to initiate the copying process. The FORCE option will not initiate a background copy.
3. After a WITHDRAW FORCE, it is likely that the target will not be in a consistent state. This is because the background copy operation might not have completed and therefore some of the target tracks will not contain the correct data. Issuing a WITHDRAW FORCE during a point-in-time instant backup will likely lead to a backup tape that is not usable.
4. If the number of remaining tracks to be copied is unknown, the value is computed as the size of the extent plus one track.
5. Due to the CP command line length limit of 240 characters, it might not be possible to specify the full 110 extents in some command entry modes.
6. FORCE is required for space-efficient targets.
7. The FORCE option does NOT initiate a background copy operation.

**Examples**

Example 1:

```
define mdisk 0200 100 100 pack01
define mdisk 0300 100 100 pack02

flashcopy establish source 200 0.60 target 300 0.50 300 70-79
Command complete: FLASHCOPY 0200 0 49 TO 0300 0 49
```

## FLASHCOPY WITHDRAW

```
Command complete: FLASHCOPY 0200 50 59 TO 0300 70 79
FLASHCOPY RELATIONSHIP 4A20BAC2 ESTABLISHED
Ready; T=0.01/0.01 10:47:25
```

```
q flashcopy hardware 5101
-----SOURCE----- -----TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
4A20BAC2 8800 5100 PACK01 100/00 5101 PACK02 100/00 0/750
4A20BAC2 8800 5100 PACK01 150/00 5101 PACK02 170/00 0/150
Ready; T=0.01/0.01 10:47:32
```

### Example 2:

To show that a hole was created in the relationship by withdrawing some of the cylinders in the middle of it:

```
flashcopy withdraw target 300 10-19
RELATIONSHIP WITHDRAWN
Ready; T=0.01/0.01 10:48:11

q flashcopy hardware 5101
-----SOURCE----- -----TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
4A20BAC2 8800 5100 PACK01 100/00 5101 PACK02 100/00 0/150
4A20BAC2 8800 5100 PACK01 120/00 5101 PACK02 120/00 0/450
4A20BAC2 8800 5100 PACK01 150/00 5101 PACK02 170/00 0/150
Ready; T=0.01/0.01 10:48:21
```

### Example 3:

To withdraw the first 40 cylinders of the target:

```
flashcopy withdraw target 300 0.40
RELATIONSHIP WITHDRAWN
Ready; T=0.01/0.01 10:48:49

q flashcopy hardware 5101
-----SOURCE----- -----TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
4A20BAC2 8800 5100 PACK01 140/00 5101 PACK02 140/00 0/150
4A20BAC2 8800 5100 PACK01 150/00 5101 PACK02 170/00 0/150
Ready; T=0.01/0.01 10:48:51
```

### Example 4:

To withdraw the entire minidisk: (Note that some of the specified cylinders did not participate in the original relationship.)

```
flashcopy withdraw target 300 0.100
RELATIONSHIP WITHDRAWN
Ready; T=0.01/0.01 10:48:58

q flashcopy hardware 5101
HCP2466E Device 5101 - No FlashCopy relationships reported.
Ready(02466); T=0.01/0.01 10:49:15
```

### Example 5:

A sample REXX script and its invocation that could be used to wait for a relationship to be withdrawn:

```
type withdraw exec
/* REXX Example for FLASHCOPY WITHDRAW */
ARG DEVNO .
DO FOREVER
CMD='FLASHCOPY WITHDRAW TARGET' DEVNO
SAY CMD
CMD
IF RC= 2469 THEN EXIT RC
'SLEEP 5 SEC'
END
Ready; T=0.01/0.01 21:18:59

flashcopy establish source 2000 target 3000
Command complete: FLASHCOPY 2000 0 1999 TO 3000 0 1999
FLASHCOPY RELATIONSHIP 20098185 ESTABLISHED
Ready; T=0.01/0.01 21:17:05
```



```

withdraw 3000
FLASHCOPY WITHDRAW failed because 30000 tracks remain to be copied to 3000.
FLASHCOPY WITHDRAW TARGET 3000
FLASHCOPY WITHDRAW failed because 24600 tracks remain to be copied to 3000.
FLASHCOPY WITHDRAW TARGET 3000
FLASHCOPY WITHDRAW failed because 19137 tracks remain to be copied to 3000.
FLASHCOPY WITHDRAW TARGET 3000
FLASHCOPY WITHDRAW failed because 13641 tracks remain to be copied to 3000.
FLASHCOPY WITHDRAW TARGET 3000
FLASHCOPY WITHDRAW failed because 8001 tracks remain to be copied to 3000.
FLASHCOPY WITHDRAW TARGET 3000
FLASHCOPY WITHDRAW failed because 2514 tracks remain to be copied to 3000.
FLASHCOPY WITHDRAW TARGET 3000
RELATIONSHIP WITHDRAWN
Ready; T=0.01/0.01 21:17:37

```

## Responses

### Response 1:

This FLASHCOPY WITHDRAW command completed successfully:

```

flashcopy withdraw target 300 0.100
RELATIONSHIP WITHDRAWN
Ready;

```

### Response 2:

This FLASHCOPY WITHDRAW command did not completed successfully:

```

q flashcopy hardware 5100
-----SOURCE----- -----TARGET-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
1FCE5D61 1800 5100 RWS001 500/00 5101 RWS002 600/00 26112/30000
Ready; T=0.01/0.01 20:11:48

flashcopy withdraw target 3000
HCPNFC2469E WITHDRAW failed because 26112 tracks remain to be copied to
3000.
Ready(02469); T=0.01/0.01 20:11:52

```

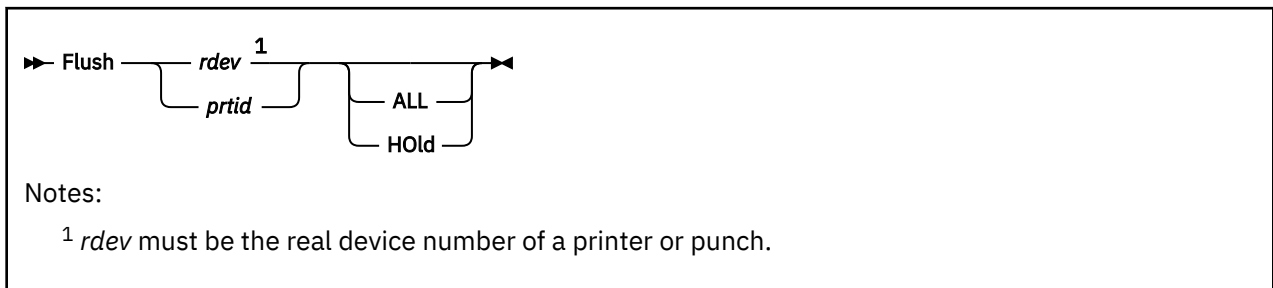
## Messages

For a complete list of messages that can be issued by the FLASHCOPY WITHDRAW command, see the [“FLASHCOPY” on page 471](#) command.

### Messages:

- HCP2469E WITHDRAW failed because *n* tracks remain to be copied to *vdev*.

## FLUSH



### Authorization

Privilege Class: D

### Purpose

Use FLUSH to do the following:

- Halt and immediately purge output on a real unit record device

It resumes activity with the next available spool file, if the unit record device meets the following conditions:

- No intervention is required.
- The device was not drained.
- The device was not taken offline.

### Operands

#### *rdev*

identifies the real device number of the real printer or punch whose current file is to be terminated. Each *rdev* must be a hexadecimal device address between X'0000' and X'FFFF'.

#### *prtId*

indicates that the entire command is to be passed for processing to the print server virtual machine designated by *prtId*. The operand *prtId* is provided by the print server virtual machine, which will enter the FLUSH command to an AFP. It is 1 to 8 alphanumeric characters long. If you specify a *prtId* of four or fewer characters, *prtId* must not contain all hexadecimal characters; because CP then interprets *prtId* as *rdev*.

#### ALL

deletes all copies of the current output file. If more than one copy of the current output file is being printed or punched and you specify ALL, all copies are deleted. If you do not specify ALL, only the current copy is deleted; and the next copy, if any, is printed or punched.

#### HOLD

places the spool file in system hold status; the file is not purged. You can reset the file status using the CHANGE command.

### Usage Notes

1. You normally use the FLUSH command when a mistake in output queuing has occurred or a high-priority job must be started before the current job has finished. For example, you would enter FLUSH if a file was directed to a printer equipped with the wrong printer train or a file was misdirected to the real punch rather than to a user's virtual reader.

2. Special handling is required if a spool file is started on a real output device that cannot be made ready. If this condition occurs, the operator receives an error message, recovers the file, and restarts it. To do so, enter **flush rdev hold**.
3. If the operator has specified HOLD, the file is placed in system hold status. If not and multiple copies of the file are to be processed, the file is requeued and started on another device, if one is available.
4. The virtual machine associated with *prt* processes the FLUSH command. Therefore, the options given on the FLUSH command must be valid for the virtual machine processing it.

## Responses

Response 1:

```
PRT rdev FILE FLUSHED userid FILE spoolid RECS nnnn COPY [*]nnn SEQ sss CL c type
PUN      FILE HELD          nnnK
        COPY FLUSHED      nnnM
```

### PRT

### PUN

identifies the type of device you have flushed.

### rdev

identifies the real device number of the printer or punch to be affected.

### FILE FLUSHED

### FILE HELD

### COPY FLUSHED

indicates the disposition for the file.

### userid

identifies the user who was to receive the output.

### spoolid

identifies the system-assigned number by which CP refers to the file.

### RECS nnnn

### RECS nnnK

### RECS nnnM

identifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines in thousands rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million.

### COPY nnn

### COPY \*nnn

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (\*) is specified, the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

If a file is not spooled to the 3800 printer, the function indicated by the asterisk (\*) is ignored and duplication occurs normally.

### SEQ sss

identifies the sequence number for the file on the printer or punch.

**Note:** CP itself does not provide a response for *prtids*. The virtual machine associated with *prt* provides the response.

### CL c

is the spool file class.

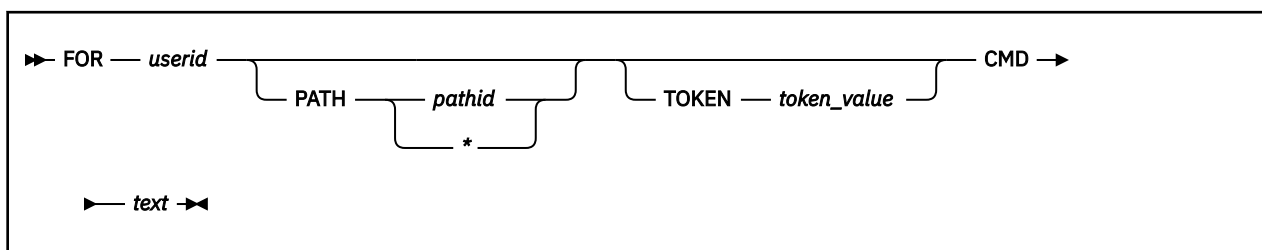
### type

shows the device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP, or SYS.

### Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP046E *type rdev* offline
- HCP140E *type {rdev/ldev}* attached to *userid*
- HCP141E *type rdev* not active
- HCP417E *{rdev/vdev}* is not an output spooling device.
- HCP418E *{rdev/vdev}* is not a spooled device.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP1043E *{Printer|Punch} rdev* was not flushed because SPXTAPE is using the file which is active on the *{printer|punch}*. Wait until SPXTAPE has finished and reissue the command.

## FOR



## Authorization

Privilege Class: C, G

## Purpose

Use FOR to execute a CP command on another virtual machine and receive the command's responses and return code either to your terminal or over an IUCV connection to the [Asynchronous CP Command Response system service \(\\*ASYNCCMD\)](#) in *z/VM: CP Programming Services*.

## Operands

### *userid*

identifies the virtual machine that is to be the target for the CP command.

### PATH *pathid*

specifies which IUCV connection to the Asynchronous CP Command Response system service should be used to receive the command's responses and return code. The *pathid* may be specified as either 1 - 4 hexadecimal digits or as an asterisk (\*). When specifying an asterisk, the most recent connection is used. When a path is not specified, command responses and the return code will be sent to the console of the FOR command issuer.

### TOKEN *token\_value*

specifies the correlation token included on each command response record and each command return code record. The *token\_value* is a 1 - 16 character, blank-delimited user-specified string. The correlation token is used to sort out interleaved responses from consecutively issued FOR commands or for any other purpose.

### CMD

indicates that the command text follows.

### *text*

specifies the CP command to issue. If *text* contains multiple commands, each command and its associated parameters must be separated from adjacent commands by the value X'15'.

## Usage Notes

1. The return code from the FOR command will be received immediately. Then, asynchronously, the CP command is executed on the target virtual machine.
2. The FOR command captures only synchronous responses from CP commands issued to the target virtual machine. Commands which generate asynchronous responses, such as SPXTAPE, REWIND, and XAUTOLOG without the SYNCH option, are not recommended for use with the FOR command. For more information, see [“Delayed Responses to CP Commands”](#) on page 7.
3. The target of the FOR command will not receive any command responses.
4. If you specify the IUCV connection to the \*ASYNCCMD system service, the format of the data sent via IUCV by the [Asynchronous CP Command Response system service \(\\*ASYNCCMD\)](#) is described in *z/VM: CP Programming Services*.

5. The format of command responses directed to the console of the FOR command issuer is as follows:

```
userid : token : text
```

where:

**userid**

is the user ID specified on the FOR command which is the target of the CP command. This field is 8 characters long and delimited by a colon with spaces on both sides.

**token**

is the correlation value specified on the FOR command. This field is 16 characters long and delimited by a colon with spaces on both sides. This field, including the delimiter and spaces, is not displayed when a token value is not specified.

**text**

is the actual response that would have displayed on the target virtual machine's console using the EMSG ON setting. The virtual processor running the CP command is not displayed as a prefix in the command response.

After all command responses are displayed to the console of the FOR command issuer, a final command return code is displayed. The *text* field will have the following format where *nnnn* is the return code value.

```
Command Complete. CP return code = nnnn
```

6. Unlike the SEND command, the FOR command does not translate the receiver's logical line-end character.
7. A class G user can issue CP commands to a virtual machine for which the user can provide secondary console services. A class C user can issue CP commands to any virtual machine.
8. The target virtual machine needs to have the correct privilege or authorization to execute the issued CP command.
9. If the FOR command is executing multiple commands and encounters an invalid command, processing stops and the remaining commands are ignored.

## Examples

Example 1:

To see a list of commands authorized for a user to issue, enter:

```
for user1 cmd commands
Ready;
USER1   : ADJUNCT  ATTN      BEGIN    CHANGE   CLOSE    COMMANDS
USER1   : COUPLE   CPU       DEFINE   DETACH   DIAL     DISCONNECT
USER1   : DISPLAY  DUMP     ECHO     EXTERNAL FOR     INDICATE
USER1   : IPL      LINK     LOADVFCB LOCATEVM
...
USER1   : DIAG278  DIAG27C  DIAG280  DIAG288  DIAG29C  DIAG2A0
USER1   : DIAG2A4
USER1   : HCPFOR069I Command Complete. CP return code = 0000.
```

Example 2:

To receive a list of commands authorized to be issued for a user with responses directed to path ID 5, an IUCV connection to \*ASYNCMD, and the token value COMMANDLIST1 included in the output, enter:

```
for user1 path 5 token COMMANDLIST1 cmd commands
```

Example 3:

To display the resources occupied by a user enter:

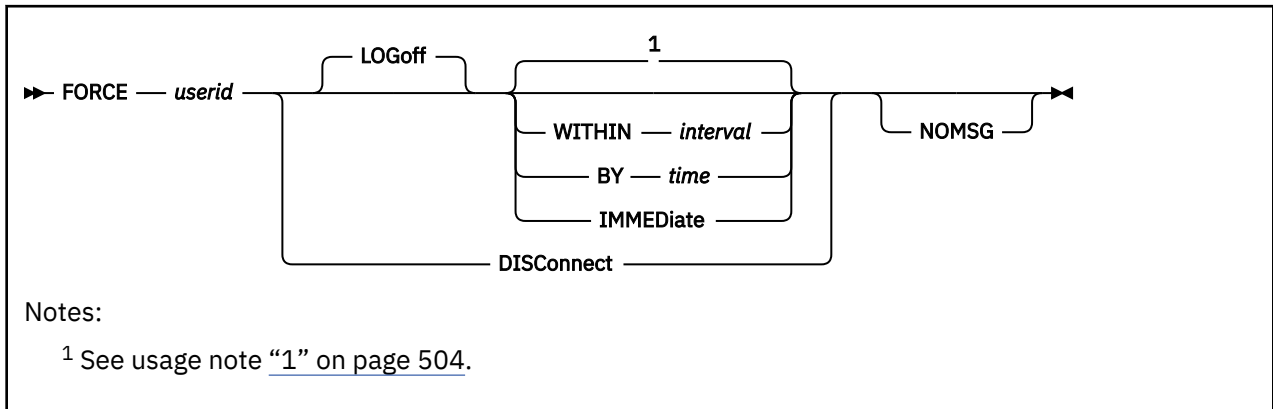
```
for user2 cmd indicate user
Ready;
USER2   : USERID=USER2  MACH=XA  STOR=128M VIRT=V XSTORE=---
```

```
USER2 : IPLSYS=CMS      DEVNUM=00016
USER2 : PAGES: RES=00001291 WS=00000625 LOCKEDREAL=00000000
USER2 : NPREF=00000000  PREF=00000000  READS=00000000  WRITES=00000000
USER2 : CPU 00: CTIME=00:03 VTIME=000:00 TTIME=000:00 IO=000533
USER2 :          RDR=000000 PRT=000000 PCH=000000
USER2 : HCPFOR069I Command Complete. CP return code = 0000.
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP069I Command Complete. CP return code = *nnnn*.
- HCP070E You are not authorized to act FOR *userid*
- HCP071E The FOR command may not be issued recursively
- HCP072E Path identifier missing or invalid
- HCP2873E Missing or invalid parameter or option.
- HCP6525E External Security Manager is unavailable.
- HCP6704E Missing token at end of line

## FORCE



### Authorization

Privilege Class: A

### Purpose

Use FORCE to disconnect or log off an active user.

### Operands

#### *userid*

identifies the user to be logged off or disconnected.

#### **LOGoff**

indicates that the user should be logged off. Logoff is the default action. When the action is logoff, you can also specify one of the WITHIN, BY, or IMMEDIATE timing operands.

#### **DISConnect**

indicates that the user should be disconnected.

#### **NOMSG**

indicates that the asynchronous FORCE response message should be suppressed. This option has no affect when issued by the primary system operator.

#### **WITHIN *interval***

sends the user a shutdown signal and delays logging them off until either the specified interval has elapsed or the user machine indicates that it has shut down, whichever occurs first. The *interval* specifies a number of seconds in the range 1 - 65535. If the user is not enabled for shutdown signals, this operand is ignored and the user is logged off immediately. See usage note “8” on page 505.

#### **BY *time***

sends the user a shutdown signal and delays logging them off until either the designated time of day is reached or the user machine indicates that it has shut down, whichever occurs first. The *time* can be specified as *hh:mm* or *hh:mm:ss*. The equivalent interval in seconds must be in the range 1 - 65535. If the user is not enabled for shutdown signals, this operand is ignored and the user is logged off immediately. See usage note “8” on page 505.

#### **IMMEDIATE**

logs off the user immediately without first sending them a shutdown signal. If the user is in the process of being logged off, any delay to allow the guest to process a shutdown signal is terminated. See usage note “8” on page 505.

### Usage Notes

- Notes about logoff timing:



- a. If the user is not enabled for shutdown signals, then the logoff begins immediately.
- b. If the user is enabled for shutdown signals, then the logoff timing is determined as follows:
  - If the user is currently in a status-pending condition, then the logoff begins immediately.
  - If the user is not in a status-pending condition, then the logoff timing is determined as follows:
    - If a timing operand (WITHIN, BY, IMMEDIATE) is specified, then the timing operand determines the logoff timing.
    - If a timing operand (WITHIN, BY, IMMEDIATE) is not specified, then the logoff is delayed by up to the amount of time that is defined by the system-default shutdown-signal timeout interval. The logoff is delayed until the system-default shutdown-signal timeout interval elapses or until the user machine indicates that it is shut down, whichever occurs first.
2. FORCE may not be able to cause the logoff of a hung-up user or a user with I/O outstanding, because either of these conditions may prevent completion of the LOGOFF function. As an alternative, the DISCONNECT option may be used to disassociate the user ID from its terminal, thus freeing up the device.
3. If the user's virtual machine is hung waiting for a device which is not operational or the system operator can determine the I/O device involved and either make that device ready or enter a HALT command; this allows the user to continue operation or be forced from the system. (Use QUERY to determine the status of all or selected devices) For more information see the description of the HALT command.
4. The disconnect performed by the FORCE command may take significant time to complete because of the attempt made to process the disconnect even in the case of a hung terminal.
5. If a user is force disconnected while a read is posted from the user's virtual machine or from CP, and that user does not have a logged on secondary user, the user then has some installation-defined grace period (usually 15 minutes) in which to log on or else the user's virtual machine is automatically logged off.
6. Users who are being forced from the system will be given QUICKDSP designation to expedite the logoff of the user.
7. If the user who is being forced from the system has an open RDR file and the user's virtual reader is spooled NOHOLD and NOKEEP, the open file will be retained on the user's RDR queue in USER hold status.
8. A user is enabled for signals only when a guest operating system is running in the virtual machine and has enabled to receive these signals.
 

If the FORCE *userid* command is specified with no operands (other than NOMSG), and the specified user is enabled for shutdown signals, and a system default shutdown signal timeout interval greater than 0 is defined (by the SET SIGNAL SHUTDOWNTIME system configuration statement or command), that timeout value is used to delay the logoff, and a shutdown signal is sent to the user to give the guest system an opportunity to do its own shutdown processing. If the system default shutdown signal timeout interval is 0, or no value is defined (which defaults to 0), or the specified user is not enabled for shutdown signals, no signal is sent, and the user is immediately logged off. Refer to [“SET SIGNAL” on page 1701](#) for additional information about this behavior.

If the specified user is enabled for shutdown signals but is currently in a status-pending condition, the user will be logged off immediately.
9. The FORCE (of the relocating guest by another user) command will fail if issued when VMRELOCATE is in progress.
10. If the forced-off user is enabled for signals, the FORCE *user* command sends a shutdown signal to the forced user and this message is sent to the primary system operator:

```
HCP2118I Shutdown signal sent to user because a FORCE was issued at
yyy-mm-dd hh:mm:ss by userid
```

If a user is enabled for shutdown signals and is currently in a status-pending condition, message HCP2118I is *not* sent to the primary operator's console when this user is forced.

## Responses

Response 1:

```
{LOGOFF } AT hh:mm:ss zone weekday mm/dd/yy BY {SYSTEM}
{DISCONNECT} {userid}
```

In addition, if LOGOFF was selected, then this response will accompany the normal accounting message produced at logoff.

### LOGOFF DISCONNECT

indicates whether the virtual machine was logged off or disconnected.

#### **hh:mm:ss**

identifies the current time in hours:minutes:seconds.

#### **zone**

identifies the time zone.

#### **weekday**

identifies the current day of the week.

#### **mm/dd/yy**

identifies the current date in month/day-of-month/year format.

### SYSTEM

#### **userid**

indicates who forced the logoff or disconnect your virtual machine. If you were forced by another user, and not the system, that user's *userid* appears in the message.

Response 2:

```
{type {rdev} } {LOGOFF AS } userid USERS = nnnnn FORCED BY {userid}
{ {ldev} } {DISCONNECT}
{luname }
```

is the message received by the primary system operator. In addition, if the Class A issuer of the command is not the primary system operator and the NOMSG option was not specified, then the command issuer also receives this response.

### LOGOFF AS DISCONNECT

indicates whether the user was logged off or disconnected.

#### **type**

identifies the display device type.

#### **rdev**

#### **ldev**

identifies the real or logical device number.

#### **luname**

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

#### **userid**

identifies the user who was forced.

#### **USERS = nnnnn**

identifies the number of users remaining on the system.

#### **userid**

### SYSTEM

indicates who forced the virtual machine. If forced by another user, and not the system, that user's *userid* appears in the message.

Response 3:

```
USER DSC LOGOFF AS userid USERS = nnnnn FORCED BY {SYSTEM} {userid}
```

is issued to the primary system operator and the command issuer (when the issuer is not the primary system operator and the NOMSG option was not specified) if the user is already disconnected, and the FORCE command is used to log that user off the system.

### **USER DSC**

indicates that the user's virtual machine was disconnected when the FORCE command was issued.

### ***userid***

identifies the user who was forced off.

### **USERS = *nnnnn***

identifies the number of users who remain on the system.

Response 4:

```
Shutdown signal may delay logoff for up to interval seconds
```

is issued to the command issuer (when the NOMSG option was not specified) if the FORCE may be delayed in order to allow the user time to shut down.

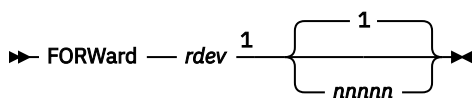
### ***interval***

is the amount of time (in seconds) that the guest will be given to terminate before it is logged off.

## **Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP013E Conflicting option - *option*
- HCP045E *userid* not logged on
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP1514E *userid* already disconnected
- HCP1801E Command *command* could not be completed due to a relocation in progress
- HCP2114I User *userid* termination may be delayed for up to *interval* seconds
- HCP2118I Shutdown signal sent to *user* because a FORCE was issued at *yyyy-mm-dd hh:mm:ss* by *userid*
- HCP6704E Missing token at end of line

## FORWARD



Notes:

<sup>1</sup> *rdev* must be the real device number of a printer or punch.

### Authorization

Privilege Class: D

### Purpose

Use FORWARD to reposition current spool output on a real printer or punch.

### Operands

***rdev***

identifies the device number of the real printer/punch you want spaced forward.

**1**

***nnnnn***

specifies the number of pages or cards to be spaced forward. The limit is 32,767; the default is 1.

### Usage Notes

Pages are counted in FORWARD operations by looking for page ejects. If a file contains no page ejects, 66 output lines (including blanks) are treated as a “page.” The count begins at the end of the current buffer.

### Responses

Response 1:

```
{PRT} rdev FORWARD userid FILE spoolid RECS {nnnn}
{PUN} {nnnK}
{nnnM}
```

```
COPY [*]nnn SEQ sss CL c typ
```

**PRT**

**PUN**

identifies the device type.

***rdev***

identifies the real device number.

***userid***

identifies the user whose file was active and spaced forward.

***spoolid***

identifies the system-assigned number of the spool file.

**RECS *nnnn*****RECS *nnnK*****RECS *nnnM***

identifies the number of logical records left in the file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines in thousands rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million.

**COPY [*\**]*nnn***

identifies the remaining number of file copies to be produced, where 001 is the last copy. The asterisk (\*) is for the 3800 printer and indicates that the printer copies each page *nnn* times before going to the next page.

**SEQ *sss***

identifies the sequence number of the printer/punch file.

**CL *c***

is the spool file class.

***typ***

shows the device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP, or SYS.

**Messages**

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*}
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP046E *type rdev* offline
- HCP140E *type {rdev/ldev}* attached to *userid*
- HCP141E *type rdev* not active
- HCP417E *{rdev/vdev}* is not an output spooling device.
- HCP418E *{rdev/vdev}* is not a spooled device.

## FREE LOGON

---

```
►► FRee — userid — LOGon ◄◄
```

### Authorization

Privilege Class: B

### Purpose

Use FREE LOGON to release a user ID from hold logon status, and allow the user to access the host system.

### Operands

#### *userid*

identifies the user to be removed from logon hold status.

#### LOGon

allows the specified user to access the host system.

### Usage Notes

1. Use FREE LOGON in conjunction with hold logon.
2. When this command is issued on a member of an SSI cluster for a user ID that is defined as a single-configuration virtual machine, the command is propagated to all the other members of the SSI cluster. If the user ID is defined as a multiconfiguration virtual machine, this command is not propagated across the SSI cluster.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP053E [XAUTOLOG failed for *userid*:] *userid/value* not in CP directory
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1553I *userid* is not in hold status for {*printer*|*punch*|*logon*|all devices}.

## FREE PRINTER/PUNCH



### Authorization

Privilege Class: D

### Purpose

Use FREE PRINTER/PUNCH to release user spool files from system hold.

### Operands

#### *userid*

identifies the user whose spool files are to be released. The user need not be logged on.

#### **ALL**

specifies that all print/punch files are to be freed. This is the default.

#### **Printer**

#### **PRT**

specifies that print files are to be freed.

#### **PUnch**

#### **PCH**

specifies that punch files are to be freed.

### Usage Notes

1. Use FREE PRT/PUN in conjunction with HOLD PRT/PUN.
2. After you enter the FREE command, all new files are put into nohold status.
3. A spool file can be freed from either or both of two types of hold status:
  - User
  - System.

All files in user hold status can be freed by the user who created the file or by a class D user.

All files in system hold status can be freed by the class D user only; the commands used are FREE and CHANGE NOSYS.

A spool file in both types of hold status must be freed from each one before the file can be processed.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid

## FREE PRINTER/PUNCH

- HCP026E Operand missing or invalid
- HCP053E [XAUTOLOG failed for *userid*:] *userid/value* not in CP directory
- HCP475I Fatal I/O error trying to read directory from *volid* [for user *userid*]
- HCP1553I *userid* is not in hold status for {printer|punch|logon|all devices}.



## GIVE

---

### Authorization

Privilege Class: B

### Purpose

Use GIVE to transfer control either permanently or temporarily to a:

- Dedicated virtual tape drive from *your virtual machine* to another user's virtual machine.
- Dedicated real tape drive from *another user's virtual machine* to a third user's virtual machine.

### Usage Notes

(General for GIVE)

1. SET IMSG OFF suppresses informational messages for this command.
2. If you enter a command to transfer a real tape drive from your virtual machine, it is still treated as transferring the virtual tape drive. In such a case, if you have different device numbers for the real device and the virtual device associated with the tape drive, the responses refer to the virtual device number, even though you specified the real device number in your command.
3. The receiver and the giver in a command cannot be the same user.
4. The GIVE command will not alter the ASSIGN/UNASSIGN state of the device.
5. Any encryption settings that were defined to the device via SET RDEV and/or ATTACH are retained across a GIVE command.

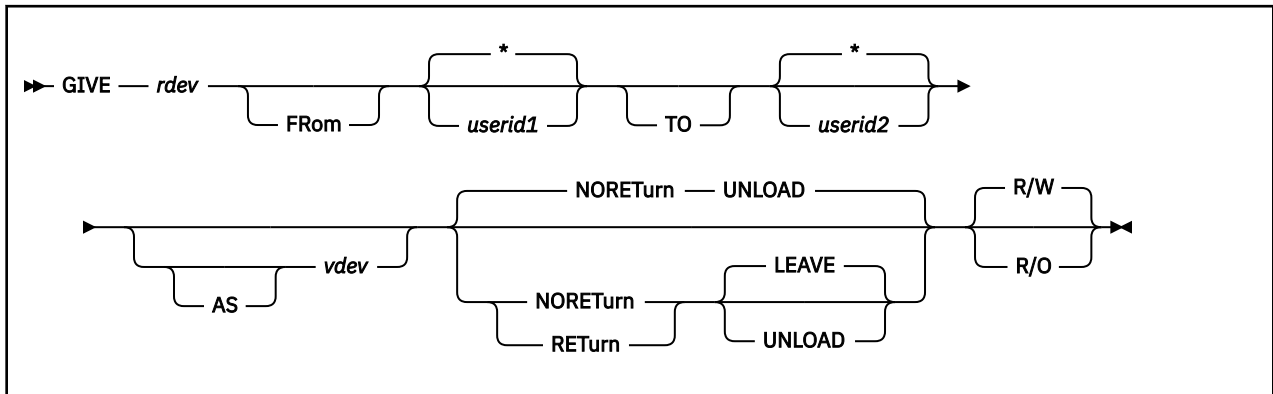
### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP006E Invalid device type - *device*
- HCP007E Invalid userid - *operand*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP021E A device number was not supplied or it was invalid.
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev|vdev*} does not exist
- HCP045E *userid* not logged on
- HCP120E *type* {*rdev|vdev*} not given.; *userid vdev* already defined
- HCP121E *type rdev* not attached to *userid*
- HCP153E Device *vdev* exceeds maximum number of virtual devices
- HCP422E The same option was specified twice.
- HCP1105E Tape *rdev* not attached; Tape assigned elsewhere.
- HCP1107E Attach failed; I/O error on tape *rdev*
- HCP1115E Device [*vdev|rdev*] was not given. A [GIVE|return] is (already) pending for device [*vdev|rdev*].
- HCP1116E Tape *rdev* not attached; the tape is being rewind.

## GIVE

- HCP1118E Device [*vdev*|*rdev*] was not given. GIVE cannot be performed on a device attached MULTIUSER.
- HCP1120E The requested GIVE for device *rdev* did not complete in the allotted time.
- HCP1123E Multiple options are possible for operand *operand*. Resolution is impossible.
- HCP1128E Device *rdev* not attached; *user* is not authorized.
- HCP1561I Real device *rdev* forced R/O; unauthorized for R/W.
- HCP6863E *userid*: Device *rdev* has not been attached because it is a DPS=NO device and the CHPIDVirtualization option is specified for the user in the directory.

## GIVE (Real Device)



### Authorization

Privilege Class: B

### Purpose

Use GIVE (Real Device) to transfer control of a real tape drive dedicated to one user, from that user's virtual machine to another user's virtual machine.

### Operands

#### *rdev*

is a real device number of the tape drive to be transferred from the indicated user.

#### *userid1*

#### **FRom *userid1***

\*

identifies the virtual machine to which the device is currently attached. The specified user must be logged on and not in the process of logging off. If you are issuing the GIVE (Real Device) command to give a device from your own virtual machine, you can specify *userid1* as an asterisk (\*).

Use the keyword FROM when:

- You specify a user ID with four numbers or fewer
- You specify a user ID that could also be a valid hexadecimal device address (for example, ABC)
- You specify *userid1* as "to" or "fr" or "fro" or "from".

#### *userid2*

#### **TO *userid2***

\*

identifies the virtual machine that receives the tape drive. If you are issuing the GIVE (Real Device) command to transfer the ownership from another user to your virtual machine, you can specify *userid* as an asterisk (\*). The receiver must be logged on to the system. The specified user ID must not be that of the giver (the virtual machine to which the device is currently attached). If the user ID of the specified receiver is also a valid hexadecimal number of not more than 4 digits and the immediate operand following this operand is not "[AS] *vdev*", the command is rejected with error message HCP1123E.

Use the keyword TO when:

- You specify a user ID with four numbers or fewer
- You specify a user ID that could also be a valid hexadecimal device address (for example, ABC)

## GIVE (Real Device)

- You specify *userid2* as "TO" or "FR" or "FRO" or "FROM".

### **vdev**

#### **AS vdev**

is the virtual number to be assigned to the tape drive when it is attached to the target virtual machine. If this is omitted, the real device number is assigned as the virtual device number. If *vdev* is specified, it should be the last operand before the options.

Use the keyword AS when device number to be specified can also be a valid user ID.

## Options

Options may be specified in any order. Conflicting options (for example, RETurn/NORETurn) cause an error message to be issued and the command to be terminated. The only valid options are:

### **RETurn**

specifies that the invoker wishes to regain control of the transferred tape drive when detached by the receiver. If this is the case, RETurn must be explicitly specified.

### **NORETurn**

specifies that the invoker does not wish to regain control of the transferred tape drive when detached by the receiver. NORETurn is the default and need not be explicitly specified.

### **R/O**

causes the tape drive to be attached to the target virtual machine in read only mode. Attempting to write to a tape drive given with R/O specified results in an error.

### **R/W**

causes the tape drive to be attached to the target virtual machine in read/write mode. R/W is the default and need not be explicitly specified.

### **LEAVE**

specifies that the tape is to be left in its position when it is detached and returned to the original owner. LEAVE is the default when RETurn is specified.

### **UNLOAD**

specifies that the tape is to be rewound and unloaded when it is detached from the receiver.

## Usage Notes

1. If a class B user issues the DETACH command to detach a GIVEN tape drive, any UNLOAD or LEAVE options specified on the DETACH command override the option that was specified on the GIVE command. If no option was specifically requested on the DETACH command, the option specified on GIVE is in effect. If LEAVE or UNLOAD are not specified on either the GIVE or DETACH command, the tape is rewound and unloaded.
2. "FRom *userid1*" and "TO *userid2*" need not be specified in the order shown if "FRom" and "TO" are used.
3. If "FROM" "TO" and "AS" are omitted, the first valid operand for a user ID is taken as the GIVER and the second valid operand for a user ID is taken as the RECEIVER.
4. "FROM *userid*" and "TO *userid*" can be interchanged in the command syntax order.
5. The LEAVE option is valid only when the RETurn option is specified.

## Responses

The following are the command responses issued to the giver and receiver as a result of successful completion of the GIVE command.

Response 1:

*giver*

```
tape vdev1 given to userid vdev2 [R/O]
```

This response is issued to the invoking virtual machine to inform the user that the tape drive was successfully transferred. If read-only access was specified, R/O is appended to the response.

Response 2:

*operator*

```
tape rdev given to userid1 vdev
from userid2 by userid3 [R/O]
```

This response is issued to the primary system operator to inform the operator that ownership of the tape drive was transferred from *userid2* to *userid1* and is attached to *userid1* at virtual device number *vdev*. If read-only access was specified, R/O is appended to the response.

Response 3:

*receiver*

```
tape vdev2 attached [R/O]
```

This response is issued to the receiving virtual machine to inform the user that a tape drive is attached. If read-only access was specified on the GIVE command, the R/O is appended to the response.

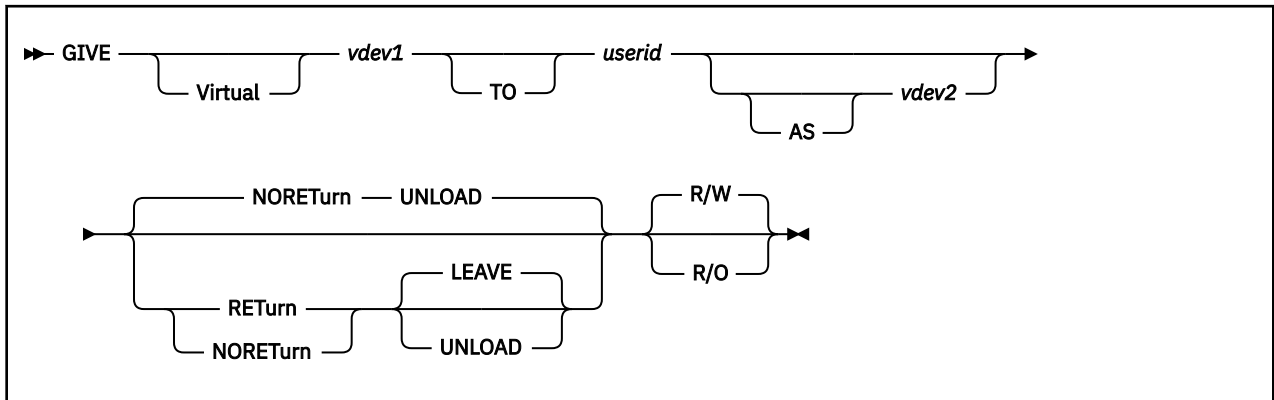
Response 4:

*originator*

```
tape rdev given to userid1 vdev
from userid2 [R/O]
```

This response is issued to the originator of the command to inform the user of the successful completion of the command. If the originator is the same as the GIVER or the operator or receiver then this response is not issued. *userid1* is the receiver of the device and *userid2* is the receiver of the device. If read-only access was specified, then R/O is appended to the response.

## GIVE (Virtual Device)



### Authorization

Privilege Class: B

### Purpose

Use GIVE (Virtual Device) to transfer control of a dedicated tape drive from your virtual machine to another virtual machine.

### Operands

#### *vdev1*

is the virtual address of the tape drive to be transferred. If this is not specified, the command is rejected.

#### *userid*

##### **TO *userid***

identifies the virtual machine that receives the tape drive. The receiver must be logged on to the system. The specified user ID must not be that of the invoking virtual machine (for example, one cannot GIVE a tape drive to oneself). If a valid user ID is not specified, the command is rejected.

Use the keyword TO when:

- You specify a user ID with four numbers or fewer
- You specify a user ID that could also be a valid hexadecimal device address (for example, ABC)
- You specify for giver or receiver a user ID that is "TO" or "FR" or "FRO" or "FROM".

#### *vdev2*

##### **AS *vdev2***

is the virtual number to be assigned to the tape drive when it is attached to the target virtual machine. If this is omitted, the real device number is assigned as the virtual device number.

Use the keyword AS when the device number to be specified can also be a valid user ID.

### Options

Options may be specified in any order. Conflicting options (for example, RETURN/NORETURN) cause an error message to be issued and the command to be terminated. The only valid options are:

#### **RETurn**

specifies that the invoker wishes to regain control of the transferred tape drive when detached by the receiver. If this is the case, RETurn must be explicitly specified.

**NORETurn**

specifies that the invoker does not wish to regain control of the transferred tape drive when detached by the receiver. NORETurn is the default and need not be explicitly specified.

**R/O**

causes the tape drive to be attached to the target virtual machine in read-only mode. Attempting to write to a tape drive given with R/O specified results in an error.

**R/W**

causes the tape drive to be attached to the target virtual machine in read/write mode. R/W is the default and need not be explicitly specified.

**LEAVE**

specifies that the tape is to be left in its position when it is detached and returned to the original owner. LEAVE is the default when RETurn is specified.

**UNLOAD**

specifies that the tape is to be unloaded and rewound when it is detached from the user. This is the default and is accepted even if RETURN is not specified.

**Usage Notes**

1. If a class B user issues the DETACH command to detach a GIVEN tape drive, any UNLOAD or LEAVE options specified on the DETACH command override the option that was specified on the GIVE command. If no option was specifically requested on the DETACH command, the option specified on GIVE is in effect.
2. The LEAVE option is valid only when the RETurn option is specified.

**Responses**

The following are the command responses issued to the giver and receiver as a result of successful completion of the GIVE command.

Response 1:

*giver*

```
tape vdev1 given to userid vdev2 [R/O]
```

This response is issued to the invoking virtual machine to inform the user that the tape drive was successfully transferred. If read-only access was specified, then R/O is appended to the response.

Response 2:

*operator*

```
tape rdev given to userid1 vdev by userid2 [R/O]
```

This response is issued to the primary system operator to inform the operator that ownership of the tape drive was transferred from *userid2* to *userid1* and is attached to *userid1* at virtual device number vdev. If read-only access was specified, R/O is appended to the response.

Response 3:

*receiver*

```
tape vdev2 attached [R/O]
```

This response is issued to the receiving virtual machine to inform the user that a tape drive is attached. If read-only access was specified on the GIVE command, the R/O is appended to the response.

# HALT

```
▶▶ HALT — rdev ◀◀
```

## Authorization

Privilege Class: A

## Purpose

Use HALT to end execution of an active channel program.



**Attention:** Use this command only in extreme cases and after careful consideration. Indiscriminate use of this command may cause unwanted results.

## Operands

*rdev*

is the device number of the real device to be stopped.

## Usage Notes

1. After issuing HALT, enter the QUERY (real devices) command to determine the status of the device. If the device requires intervention, you should make the device ready.

## Responses

Response 1:

```
HALT INITIATED TO type rdev
```

indicates that the system has accepted the command and initiated action to halt the device.

Response 2:

```
HALT NOT INITIATED BECAUSE type rdev IS NOT ACTIVE
```

indicates that a HALT was issued to an idle device and that the system has not accepted the command.

*type*

identifies the device halted. The possible *type* values are:

Type	Meaning
DASD	Direct access storage device
FCP	SCSI device adapter
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
CONS	Console
CTCA	Channel-to-channel adapter



Type	Meaning
OSA	Open Systems Adapter
CTLR	3745 Communications Controller
MSC	Mass Storage Controller
SWCH	Dynamic Switching Device
DEV	Any other device
LINE	Communication line.

**rdev**

identifies the real device.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist

## HOLD LOGON

```
►► Hold — userid — LOGon ◄◄
```

### Authorization

Privilege Class: B

### Purpose

Use HOLD LOGON to prevent a user from accessing the host system.

### Operands

#### *userid*

identifies the user you want to prevent accessing the host system. The user need not be logged on.

#### LOGon

prevents the specified user logging on the host system.

### Usage Notes

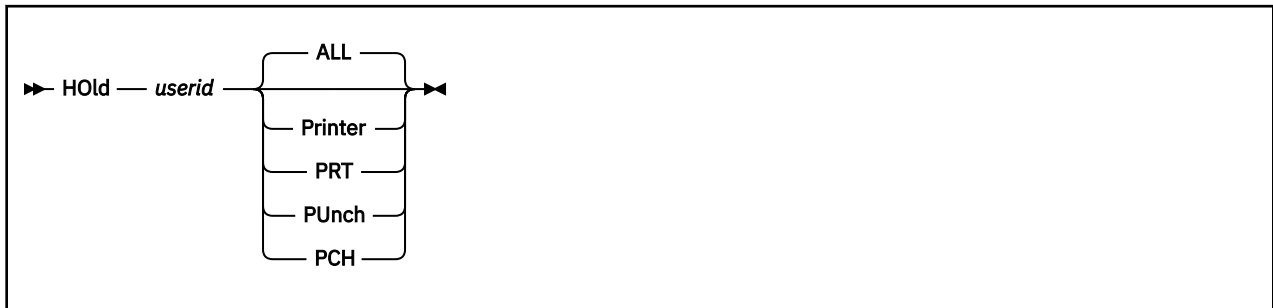
1. Use the FREE command to release a user from logon hold status.
2. The HOLD LOGON command is not allowed for the user ID defined as the primary system operator on the SYSTEM\_USERIDS statement in the system configuration file.
 

**Note:** If a primary system operator user ID is not specified, the default is OPERATOR.
3. A maximum of 32 users can be put into hold at any one time.
4. If a guest is logged onto one system in an SSI cluster, it cannot be relocated to another system that has a logon hold in effect for that user. The VMRELOCATE command will fail if this is attempted.
5. When this command is issued on a member of an SSI cluster for a user ID that is defined as a single-configuration virtual machine, the command is propagated to all the other members of the SSI cluster. If the user ID is defined as a multiconfiguration virtual machine, this command is not propagated across the SSI cluster.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid *userid* - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP053E [XAUTOLOG failed for *userid*:] *userid/value* not in CP directory
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1503E Hold status is not set because the hold queue is full.

## HOLD PRINTER/PUNCH



### Authorization

Privilege Class: D

### Purpose

Use HOLD PRINTER/PUNCH to place user spool files into hold status.

**Note:** Until you enter the FREE command, all new files are put into hold status.

### Operands

#### *userid*

identifies the user whose spool files are to be placed in system hold status. The user need not be logged on.

#### **ALL**

specifies that all printer/punch files are to be put into hold.

#### **Printer**

#### **PRT**

#### **PUnch**

#### **PCH**

specifies the type of file to be put into hold.

### Usage Notes

1. Use FREE PRT/PUN in conjunction with HOLD PRT/PUN.
2. After you enter the HOLD command, all new files are put into hold status.
3. A spool file can be put into either or both of two types of hold status:
  - User
  - System.

A spool file is put into user hold status by its creator; the commands used are SPOOL, CLOSE, or CHANGE.

A spool file is put into system hold status by the privilege class D user; the command used is CHANGE SYS. Also, the operating system can place a file into system hold status if an error occurs.

The user of the HOLD command can put the files of any user into system hold status.

A spool file in both types of hold status must be freed from each one before the file can be processed.

4. User hold status is maintained across a warm start regardless of whether the user is logged on.
5. When a user is put into hold, all current files are changed to system hold, and new files are put into hold status also.

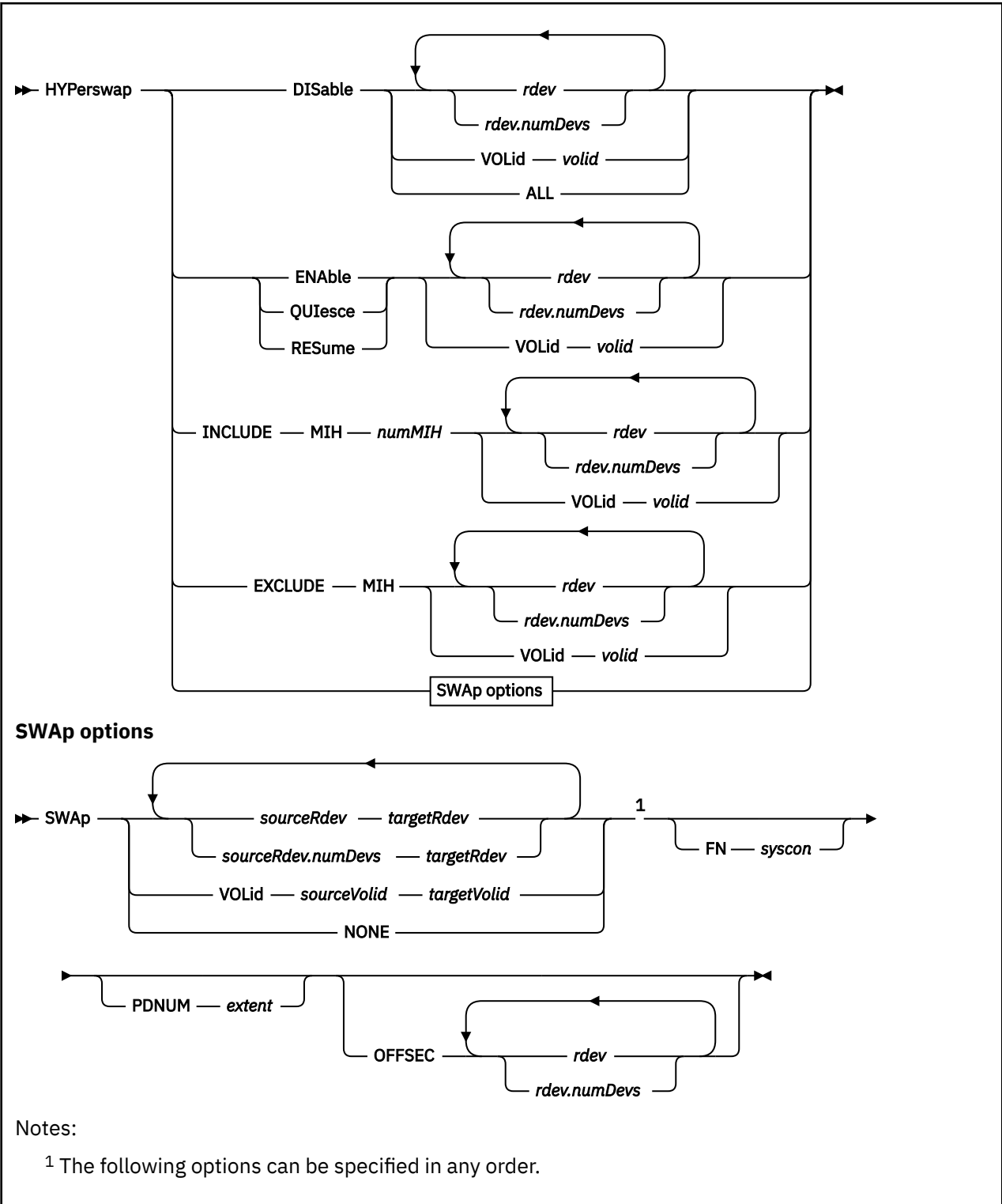
## HOLD PRINTER/PUNCH

6. Spool files in hold status are not selected for processing by a real printer or punch. Use the HOLD command to temporarily detain a file or a group of files so that you can correct or modify a spooling condition.
7. A maximum of 32 users can be put into hold at any one time.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP053E [XAUTOLOG failed for *userid*:] *userid* not in CP directory
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1503E Hold status is not set because the hold queue is full.

# HYPERSWAP



## Authorization

Privilege Class: B

## Purpose

Use the HYPERSWAP command to perform Disaster Recovery failover processing on PPRC-linked DASD.

The DISABLE function disables the real devices for HyperSwap® automatic quiesce processing.

The ENABLE function enables the real devices for HyperSwap automatic quiesce processing.

The QUIESCE function stops de-queuing all Start I/O requests hanging off the queues for the real devices entered.

The RESUME function starts de-queuing all Start I/O requests hanging off the queues for the previously quiesced devices.

The INCLUDE function indicates the devices can be automatically quiesced on a missing interrupt condition. The *numMIH* parameter specifies the number of times missing interrupt recovery will take place before automatic quiesce will take place.

The EXCLUDE function indicates a missing interrupt condition will not trigger an automatic quiesce for the specified devices.

The SWAP function will swap all relevant real device information associated with the primary Peer-to-Peer Remote Copy (PPRC) device (source) entered as the first device in the device pair with the secondary PPRC device (target) that is entered as the second device in the device pair. When operating on ranges of device pairs the range length (*numDevs*) is equal for both devices and there exists a one-to-one correspondence device number.

## Operands

### DISable

tells CP to perform the Disable function.

### ENABle

tells CP to perform the Enable function.

### QUIesce

tells CP to perform the Quiesce function.

### RESume

tells CP to perform the Resume function.

### INCLUDE

tells CP to perform the INCLUDE function.

### EXCLUDE

tells CP to perform the EXCLUDE function.

### *rdev*

#### *rdev.numDevs*

specifies a single real device (*rdev*) or a real device combined with a decimal range count (*numDevs*). A single value (*rdev*), a series (*rdev rdev*), a range (*rdev.numDevs*) or a combination of all can be entered. For a device in the active configuration, the device number entered must be a 4-digit hexadecimal number between X'0000' and X'FFFF'. For a device in the standby configuration, the device number must be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured.

### VOLid *valid*

tells CP to use a volume serial identifier. The value for *valid* is a 1- to 6-character alphanumeric identifier. Only one *valid* can be entered.

### ALL

specified on the Disable function to disable all real devices previously enabled for automatic quiesce processing.

**MIH****numMIH**

indicates whether the missing interrupt handler (MIH) is included or excluded as a trigger for automatic quiesce and *numMIH* indicates the number of times missing interrupt recovery will take place before a HyperSwap is initiated by an automatic quiesce. The *numMIH* value must be 1 to 99. The default for all DASD, which is set at IPL time, is that MIH would not trigger an automatic quiesce.

**SWAP**

tells CP to perform the SWAP function.

**sourceRdev targetRdev****sourceRdev.numDevs targetRdev**

specifies a single source real device (*sourceRdev*) and target real device (*targetRdev*) pair or source real device combined with a decimal range count (*numDevs*) and target real device range pair. A single source/target pair, a series of source/target pairs (*sourceRdev targetRdev sourceRdev targetRdev*), a range of source/target pairs (*sourceRdev.numDevs targetRdev*) or a combination of all can be entered. The devices entered must be hexadecimal real device numbers between X'0000' and X'FFFF' or, for target devices, a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured.

**VOLid sourceVolid targetVolid**

specifies a source volume serial identifier (*sourceVolid*) and target volume serial identifier (*targetVolid*) pair. Only one source/target pair can be entered.

**NONE**

specifies that no real devices will be swapped by the command. This operand should be used with the OFFSEC operand in particular scenarios where a large number of additional secondary devices need to be varied offline.

**FN syscon**

identifies to CP the file name of the new system configuration file that CP will use after a shutdown re-IPL or software restart. The file type of the new system configuration file must be CONFIG. The value of *syscon* can be 1 to 8 characters.

**PDNUM extent**

identifies the extent number of the PARM disk that CP will use after a shutdown re-IPL or software restart. The value of *extent* can be an integer between 1 and 32767.

**OFFSEC**

tells CP to vary offline specified additional secondary devices in a Multi-Target PPRC scenario.

**Usage Notes**

1. All real devices entered on the command must be DASD devices and a DASD type of 3390.
2. The ENABLE function requires an Escon or Ficon attached ECKD IBM DASD storage controller in support of 'PPRC' failover processing.
3. The SWAP function requires the state of the devices being swapped be either quiesced, offline or online, and free. An attempt will be made to VARY ON any offline devices.
4. The SWAP function requires that the device pair be the same DASD type, have matching subclass information, be free of any active I/O and the target device's size be at least as big as the source devices.
5. When the SWAP function completes successfully for a device pair, the source and target devices are left in a quiesce state. The source device is also placed in an offline state.
6. The RESUME function will only complete successfully on devices that are quiesced.
7. Error messages will be issued for individual devices and device pairs for which the command processing fails, but the command processing continues for those devices that don't fail.
8. Once the real devices in a PPRC pair are enabled for automatic quiesce processing, the devices can be automatically quiesced on a missing interrupt condition for the source device. The following steps must be taken in order to enable automatic quiesce processing:

- a. The source device must be enabled for HyperSwap.
- b. The source device must be set to check for missing interrupts, which can be enabled via the SET MITIME command.
- c. The source device must be enabled for automatic quiesce on a missing interrupt condition using the HYPERSWAP INCLUDE command.

Once triggered, the automatic quiesce process creates an error record that can be used by the Geographically Dispersed Parallel Sysplex (GDPS<sup>®</sup>) service to trigger a swap between the two devices.

9. The SWAP function will transfer a real reserve held on the source device for fullpack minidisks. The real reserve state is tracked by VM for fullpack minidisks on DASD marked SHARED by an RDEV statement or SET RDEV command and for all DASD in an SSI cluster. Virtual reserves are also maintained as part of the SWAP command. Note that there is no VM software tracking of reserve state for dedicated DASD. Reserves on such devices will be lost following a HyperSwap, resulting in a data integrity exposure. You can, however, use dedicated DASD that do not use reserve/release. Dedicated DASD using reserve/release must not be used in a GDPS environment that is enabled for HyperSwap.
10. HYPERSWAP ENABLE, INCLUDE, QUIESCE, and SWAP are not allowed for a static Parallel Access Volume (PAV) or HyperPAV base device with a guest dedicated or guest exploited alias configuration. For a base PAV, this means the specified RDEV/VOLID is either dedicated to a guest with at least one alias associated with the base PAV also dedicated to the guest or is system attached and has one or more alias PAVs defined using DEFINE PAVALIAS, DASDOPT PAVALIAS, or MINIOPT PAVALIAS. For a base HyperPAV, this means the RDEV/VOLID is either dedicated to a guest with at least one alias associated with the base HyperPAV also dedicated to the guest or is system attached and has one or more alias HyperPAVs defined using DEFINE HYPERPAVALIAS which reside in the same pool as the specified base HyperPAV.
11. The RDEV/VOLID specified on the HYPERSWAP command cannot be an alias HyperPAV or alias PAV device.
12. When used by an IBM Geographically Dispersed Parallel Sysplex<sup>®</sup> (IBM GDPS) offering, the HYPERSWAP command is not supported for real devices accessed by z/OS guests, either as dedicated devices or minidisks. Environments other than IBM GDPS have not been tested.
13. SSI members running as second-level guests of an underlying z/VM partition are not supported when HYPERSWAP is used on the underlying first-level z/VM system.
14. GDPS does not support HYPERSWAP issued from second-level z/VM systems.
15. When the SWAP function is used to swap primary and secondary devices in different subchannel sets, the low-order 4 digits of the source and target device numbers must be identical. The high-order digit, if specified, is the subchannel set where the device is configured.
16. If a HYPERSWAP ENABLE is performed on a PPRC secondary device, an attempt is made to SWAP that device with its counterpart (same device number in the other subchannel set). If the SWAP is successful, the PPRC primary is enabled. If the SWAP fails, it reports an error and the ENABLE is not performed.
17. A single Multi-Target PPRC configuration requires at least three devices. Each pair of these three devices has the potential to initiate a SWAP function.
18. As part of HyperPAV or PAV alias device processing during hyperswap, when a SWAP is done between subchannel sets, alias devices associated with the target base device are brought into the active RDEV configuration. This allows those alias device to then be referenced directly with 4 digit RDEVs following the completion of the SWAP.

## Responses

Response 1:

When you issue the HYPERSWAP command and an option, you will receive one of the following responses:

```
{Disable} processing completed
{Enable }
{Quiesce}
```



```
{Resume }
{Include}
{Exclude}
```

Response 2:

When you issue the HYPERSWAP QUIESCE or RESUME command, the system operator will receive one of the following responses respectively:

```
Hyperswap processing {beginning}
                    {completed}
```

## Examples

Example 1:

To QUIESCE real device 500 enter the following:

```
HYP QUI 500
```

Example 2:

To QUIESCE the device with a volume serial id of MNT500 enter the following:

```
HYP QUI VOL MNT500
```

Example 3:

To QUIESCE real devices 500 and 502 through 504 enter the following:

```
HYP QUI 500 502.3
```

Example 4:

To SWAP real device 500 with real device 505 enter the following:

```
HYP SWA 500 505
```

Example 5:

To SWAP real device 500 with real device 505 and 501 with 506 enter the following:

```
HYP SWA 500.2 505
```

Example 6:

To SWAP the device with a volume serial ID of MNT500 with a device that has volume serial ID MNT505 enter the following:

```
HYP SWA VOL MNT500 MNT505
```

Example 7:

To RESUME real devices 500 through 503 enter the following:

```
HYP RESU 500.4
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6407E *{rdev|valid}* cannot be swapped.
- HCP6408E *{rdev|valid}* has I/O active for processing.
- HCP6409E *{rdev|valid}* and *{rdev|valid}* are different DASD types.

## HYPERSWAP

- HCP6410E The number of extents on *{rdev|volid}* exceeds that of *{rdev|volid}*.
- HCP6411E *{rdev|volid}* is not quiesced.
- HCP6412E *{rdev|volid}* is not a valid DASD type.
- HCP6413E Subclass information for *{rdev|volid}* and *{rdev|volid}* does not match.
- HCP6414E *{rdev|volid}* and *{rdev|volid}* cannot be swapped because the target device's VOLID cannot be read or written.
- HCP6415E *{rdev|volid}* and *{rdev|volid}* cannot be swapped because the subchannels for the device pair cannot be modified.
- HCP6416E *{rdev|volid}* is already in the process of being swapped.
- HCP6417E System detected I/O errors were encountered while attempting to release the real reserve on source *{rdev|volid}*.
- HCP6418E System detected I/O errors were encountered while attempting to obtain the real reserve on target *{rdev|volid}*. The real reserve held on the source RDEV was dropped.
- HCP6420E Target *{rdev|volid}* is holding the real reserve.
- HCP6421E *{rdev|volid}* cannot be swapped with itself.
- HCP6428I *{rdev|volid}* is an alias HyperPAV device and is not valid for the HYPERSWAP command.
- HCP6428I *{rdev|volid}* is an alias PAV device and is not valid for the HYPERSWAP command.
- HCP6429E *{rdev|volid}* is part of an unsupported configuration for the HYPERSWAP command.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## INDICATE

---

### Purpose

Use the INDICATE commands to display at your console the use of, and contention for, system resources.

If you are a class G user, use the INDICATE command to:

1. Display recent contention for system resources. This can be helpful to predict system throughput and response time characteristics that your virtual machine may experience now and in the near future.
2. Display environment characteristics of your virtual machine. This includes machine type, the origin of the system IPLed (loaded) in your virtual machine, and the presence or quantity of system resources available to your virtual machine.
3. Display measurements of resources used by your virtual machine. Such measurements are accumulators—that is, always increasing after virtual machine logon.

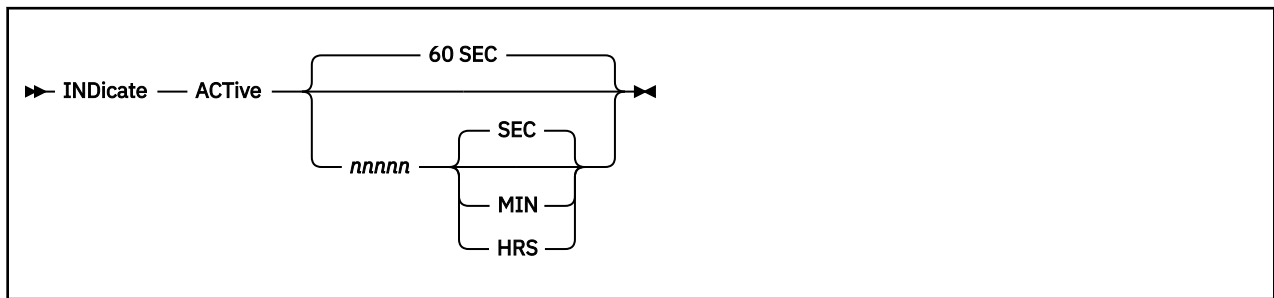
If you are a class E user (and, in some cases, a class B user), the INDICATE command provides all class G functions and the following also:

1. Detailed information on use of, and contention for, system resources. User IDs of virtual machines currently using certain resources can be displayed.
2. The status of currently active virtual machines as determined by the system scheduler and dispatcher.
3. Environment characteristics of, and measurements of resources used by, any virtual machine logged on.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP1001E An operand is missing for *option*
- HCP1006E The command processing cannot complete because system resources are not available.
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.
- HCP6005E Option *option* is not supported by z/VM.

## INDICATE ACTIVE



### Authorization

Privilege Class: E

### Purpose

Use INDICATE ACTIVE to do the following:

- Display the total number of users active in a specified time interval.
- Display the number of users in the dispatch, eligible, and dormant lists that were active in a specified time interval.

### Operands

#### *nnnnn*

identifies an integer number specifying the time interval. The valid time period is 0 to 9 hours or 540 minutes or 32400 seconds. The default is 60 seconds.

#### **SEC**

identifies that *nnnnn* was specified in seconds. This is the default.

#### **MIN**

identifies that *nnnnn* was specified in minutes.

#### **HRS**

identifies that *nnnnn* was specified in hours.

### Usage Notes

1. The number of active users indicated by the response is the actual number of users that were active in the period, not an averaged number.
2. CP's virtual processor management has been improved so that no user stays in the eligible list more than an instant before being added to the dispatch list. Therefore users are rarely shown to be in the eligible list, and the count of users in the eligible list is usually 0.

### Responses

Response 1:

```
tttt USERS, aaaa DISP, bbbb ELIG, cccc DORM
```

#### **tttt**

the total number of users that were active in the specified time interval.

#### **aaaa**

the total number of users that were active in the specified time interval, currently in the dispatch list.

**bbbb**

the total number of users that were active in the specified time interval, currently in the eligible list.

**cccc**

the total number of users that were active in the specified time interval, currently in the dormant list.

**Messages**

- HCP002E Invalid operand - *operand*

## INDICATE I/O

---

►► INDicate — I/O ◄◄

### Authorization

Privilege Class: B, E

### Purpose

Use INDICATE I/O to do the following:

- List all virtual machines currently in an I/O wait state
- List, by virtual machine, the real device number to which the most recent virtual I/O operation was mapped.

### Usage Notes

1. Because the response indicates only an instantaneous sample, enter the INDICATE I/O command several times before you assume a condition to be persistent.

### Responses

Response 1:

```
userid1 xxxx userid2 yyyy...
(as many as five users per line)
```

#### **userid**

identifies the user ID of the virtual machine.

#### **xxxx**

#### **yyyy**

identifies the real device to which the most recent I/O operation was mapped. When the I/O device cannot be determined, four hyphens (----) are displayed.

#### **Note:**

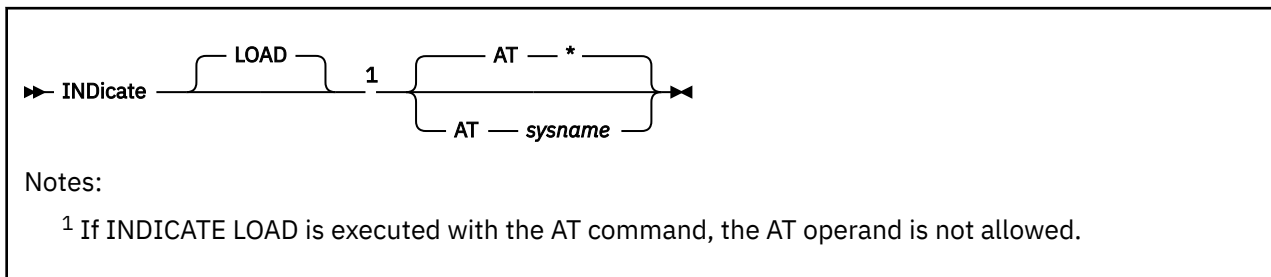
1. Where a virtual machine might have issued multiple I/O operations, the response indicates the real device corresponding to the most recent operation issued.
2. Guest I/O operations that are initiated on a PAV or HyperPAV alias minidisk will appear as I/O waiting on the appropriate real base device.

Response 2:

```
NO USERS IN I/O WAIT
```

is issued when appropriate.

## INDICATE LOAD



### Authorization

Privilege Class: E, G

### Purpose

Use INDICATE LOAD to display the operating load on the host system:

- The percentage of usage and CPU type for each online processor in your system
- Information concerning the minidisk cache
- The paging rate
- The number of users in the dispatch, eligible, and dormant lists.

**Note:** LOAD is the default operand of the INDICATE command.

### Operands

#### AT \*

displays operating load information for the system where you are logged on.

#### AT *sysname*

displays operating load information for a specific system in the SSI cluster to which this system belongs.

### Usage Notes

1. The term user(s) for the INDICATE LOAD command does not map one to one with the number of logged on users but instead maps to the number of entities that are dispatched by the dispatcher. In addition to logged on users, the dispatcher also dispatches each user's additional virtual CPUs, logon skeleton users, and other pseudo users.
2. The values listed as rates or utilizations are actually smoothed averages. At certain times, these values might differ from the values obtained from other sources (such as Performance Toolkit for z/VM), which use different algorithms.
3. A logical partition in which z/VM is running can polarized either horizontally or vertically. When a partition is horizontally polarized, all CPUs in the partition (within a given CPU type) receive access to approximately equal real CPU resources on the machine. When a partition is vertically polarized, CPUs in the partition receive different real CPU entitlements. High entitlement allows a CPU to consume approximately an entire real CPU. Medium entitlement CPUs are entitled to some percentage of a real CPU. Low entitlement CPUs receive no entitlement to real CPU resources and can consume only spare CPU resources that become available when a logical partition on the machine does not consume its entitlement. Utilization values might be lower on medium or low entitlement CPUs than on high entitlement CPUs of the same type. The utilization levels will depend on whether there are spare cycles available for consumption and whether the z/VM system requires the cycles. To change the polarization of the logical partition, use the SET SRM command with the POLARIZATION operand.

## INDICATE LOAD

4. When multithreading is enabled, use the INDICATE MULTITHREAD command to display processor core utilization information.
5. CP's virtual processor management has been improved so that no user stays in the eligible list more than an instant before being added to the dispatch list. Therefore users are rarely shown to be in the eligible list, and the count of users in the eligible list is usually 0.

## Responses

Response 1:

The class G response appears as:

```
AVGPROC-nnn% pppp
MDC READS-n..n/SEC WRITES-m..m/SEC HIT-RATIO-hhh%
PAGING-n..n/SEC
Q0-nnnnn Q1-nnnnn          Q2-nnnnn EXPAN-nnn Q3-nnnnn EXPAN-nnn
```

Response 2:

The class E response appears as:

```
AVGPROC-nnn% pppp
MDC READS-n..n/SEC WRITES-m..m/SEC HIT-RATIO-hhh%
PAGING-n..n/SEC
Q0-nnnnn (zzzzz)          DORMANT-nnnnn
Q1-nnnnn (zzzzz)          E1-nnnnn (zzzzz)
Q2-nnnnn (zzzzz) EXPAN-nnn E2-nnnnn (zzzzz)
Q3-nnnnn (zzzzz) EXPAN-nnn E3-nnnnn (zzzzz)

PROC adr1-nnn% {CP } pe   PROC adr2-nnn% {CP } pe
                {IFL }
                {ZIIP}
                {ICF }
PROC adrn-nnn% {CP } pe   PROC adrn-nnn% {CP } pe
                {IFL }
                {ZIIP}
                {ICF }
:
LIMITED-mnnnn
```

### AVGPROC-*nnn*%

identifies the average processor use percentage for all processors combined. The *nnn* value is 0 through 100 percent. This value represents an average value of the portion of a real CPU that each logical CPU was able to consume.

### *pppp*

identifies the number of processors.

### READS-*n..n* /SEC

identifies the number of data pages read from the minidisk cache instead of from DASD. The *n..n* value will be displayed as 6 - 8 digits.

### HIT RATIO-*hhh*%

identifies the percentage of successful lookups in the minidisk cache. This is the percentage of eligible read I/Os that were avoided because of the minidisk cache.

### WRITES-*m..m* /SEC

identifies the number of data pages inserted into the minidisk cache. The *m..m* value will be displayed as 6 - 8 digits.

### PAGING-*n..n* /SEC

identifies a value that indicates the current system paging activity in decimal, including migration writes, but does not include spooling information. The *n..n* value will be displayed as 1 - 8 digits.

### Qn

indicates a smoothed average of the number of Qn users in the dispatch list. A Qn user is an En user who is now in the dispatch list.



**Q0**

indicates the users that were added to the dispatch list with no delay in the eligible list, regardless of the length of their current transaction.

**Q1**

indicates the users that have just begun a transaction, and therefore are assumed to be currently processing short transactions.

**Q2**

indicates the users that did not complete their current transactions during their first dispatch list stay and therefore are assumed to be running medium-length transactions.

**Q3**

indicates the users that did not complete their current transactions during their second dispatch stay and therefore are assumed to be running long transactions.

**EXPAN-*nnn***

indicates the total delay in response time that a virtual machine experiences because of contention for scheduled resources. *Q2* gives the expansion factor for those completing in *Q2* (medium-length), and *Q3* for those completing in *Q3* (long-running transactions).

**En**

indicates a smoothed average of the number of *En* users in the eligible list. Although a user may briefly be considered an *E0* user, no *E0* counts appear. This is because, by definition, such a user spends virtually no time in the eligible list. After smoothing, this count would always be 0 if shown. If, by unusual timing circumstances, a user is found in an *E0* category, the user is counted with the *Q0* user counts.

**E1**

indicates the users that have just begun a transaction and therefore are assumed to be currently processing short transactions.

**E2**

indicates the users that did not complete their current transactions during their first dispatch list stay and therefore are assumed to be running medium-length transactions.

**E3**

indicates the users that did not complete their current transaction during their second dispatch list stay and therefore are assumed to be running long transactions.

**ZZZZ**

identifies the number of users within the *Qn* or *En* counts that are designated as *loading users*. A loading user is a user assumed to be loading a working set into storage. A user is labeled loading or not loading based primarily on the number of page reads done during the user's last dispatch time slice. A high count of page reads is assumed to designate a loading user. A user also may be labeled as loading if the system has moved any of the user's referenced pages out to DASD while the user was not in the dispatch list. The loading designation is given in anticipation of the user's need to read pages back into storage when next dispatched.

**DORMANT-*nnnnn***

identifies a smoothed average of the number of dormant list users.

**PROC *adrn-*nnn**%**

represents the percentage of time that the system is running on the logical processor indicated. The operand *adrn* represents the processor ID in hexadecimal. The *nnn* value is 0 through 100 percent. This value represents the portion of a real CPU that this logical CPU was able to consume.

**CP**

indicates the CPU is a central processor (that is, a general-purpose processor).

**IFL**

indicates the CPU is an IBM Integrated Facility for Linux processor.

**ZIIP**

indicates the CPU is an IBM z Integrated Information Processor.

## INDICATE LOAD

### **ICF**

indicates the Internal Coupling Facility processor.

### **pe**

indicates the polarization and entitlement of a CPU.

### **HZ**

indicates the CPU is horizontally polarized.

### **VH**

indicates the CPU is vertically polarized with high entitlement.

### **VM**

indicates the CPU is vertically polarized with medium entitlement.

### **VL**

indicates the CPU is vertically polarized with low entitlement.

### **LIMITED-*mmmmm***

is the instantaneous number of users in the limit list. These users are also in the dispatch list.

## Messages

- HCP1142E The AT option is not allowed when this command is executed remotely with the AT command.

## INDICATE MULTITHREAD



### Authorization

Privilege Class: E

### Purpose

Use INDICATE MULTITHREAD to display the system's multithreading (MT) status and processor core utilization information.

### Usage Notes

1. For information about configuring multithreading, see the [Multithreading Statement](#) in *z/VM: CP Planning and Administration*. To display thread information, use the QUERY MULTITHREAD command.  
To modify the multithreading configuration, use the [“SET MULTITHREAD”](#) on [page 1578](#)
2. The values listed as rates or utilization are averaged over the interval time. These values might differ from the values obtained from other sources (such as Performance Toolkit for z/VM), which use different algorithms.
3. The values for productivity, utilization, capacity factor, and maximum capacity factor are rounded to the nearest whole number. More precise numbers and core run times are available from sample monitor records.
4. The interval time is listed on the second line of output when multithreading is enabled. Usually this interval time will be 30 seconds; however, certain long running tasks (for example, SNAPDUMP) might cause the interval time to be longer.

### Responses

Response 1:

```
The Multithreading Facility is not installed.
```

is displayed if the multithreading facility is not installed on the hardware or the logical partition, or if the z/VM system is running as a guest.

Response 2:

```
Multithreading is disabled.
```

is displayed if the processor or configuration supports multithreading but multithreading is disabled. There are a number of reasons why multithreading might be disabled. For example, a MULTITHREADING statement might not be included in the system configuration file. Review the system initialization messages in the operator's console log.

Response 3:

```
Multithreading is enabled.  
An error occurred in the HISMT service. Use of the service is disabled.
```

is displayed if the processor or configuration supports multithreading but an error occurred while setting up the HISMT service to collect multithreading statistics. At the time of the error, an MTC001 soft abend should have been generated. Collect the dump and contact your IBM Support Center personnel.

Response 4:

```
Multithreading is enabled.
Statistics from the interval hh:mm:ss - hh:mm:ss
Core Type xxxx Busy bbb% TD ee.ee of tt Prod ddd% Util ggg%
CF ffff% MaxCF mmmm%
Core Type xxxx Busy bbb% TD ee.ee of tt Prod ddd% Util ggg%
CF ffff% MaxCF mmmm%
:
```

is displayed if the processor or configuration supports multithreading and multithreading is enabled in the system configuration file. A Core Type section is displayed if a core of that CPU type is online in the configuration.

**hh:mm:ss**

is the time stamp for the start and end times of the statistics interval shown in this response. Leading zeros are not removed.

**Core Type xxxx**

indicates the start of a section of information about cores of a certain CPU type. A Core Type section is displayed for each CPU type with at least one processor online in the configuration. The core type xxxx is one of the following:

**CP**

indicates this section pertains to central processor (that is, general-purpose processor) cores.

**IFL**

indicates this section pertains to IBM Integrated Facility for Linux processor cores.

**ZIIP**

indicates this section pertains to IBM z Integrated Information Processor cores.

**ICF**

indicates this section pertains to Internal Coupling Facility processor cores.

**Busy bbb%**

indicates the multithreading core busy time for cores of a certain CPU type. Multithreading core busy time is the percentage of the interval time that the core was in use. The value bbb is a decimal number 0 - 100 with leading zeros removed. If all cores of this type were recently varied online, or if there was an error collecting data, the value will be "---".

**TD ee.ee**

indicates the average of the multithreading core thread density for cores of a certain CPU type. Multithreading core thread density is the average number of threads in use while the core was in use. Periods of time when no thread was in use are ignored. The value ee.ee is dependent on the number of threads per core. Leading zeros are removed. If all cores of this type were recently varied online, or if there was an error collecting data, the value will be "-----".

**of tt**

indicates the number of online (activated) threads per core for this CPU type. This number is based on the available hardware, software, and information from the system configuration file. The value tt is a 1- to 2-digit decimal number with leading zeros removed. For the definition of activated threads, see the explanation of the "Activated Threads" field in the response for the QUERY MULTITHREAD command.

**Prod ddd%**

indicates the multithreading core productivity for cores of a certain CPU type. Multithreading core productivity is the percentage of the maximum core capacity that was used while the logical core was dispatched on physical hardware. The value ddd is a decimal number 0 - 100 with leading zeros removed. If the value is 100, all threads on the core are executing work and all core resources are being used. If all cores of this type were recently varied online, or if there was an error collecting data, the value will be "---".

**Util ggg%**

indicates the multithreading core utilization for cores of a certain CPU type. Multithreading core utilization is the percentage of the maximum core capacity that was used during the interval. The

value *ggg* is a decimal number 0 - 100 with leading zeros removed. If all cores of this type were recently varied online, or if there was an error collecting data, the value will be "----".

### CF *ffff*%

indicates the multithreading capacity factor (CF) for cores of a certain CPU type. The multithreading capacity factor is an estimate of the ratio of the amount of work that has been accomplished to the amount of work that would have been accomplished if multithreading was disabled. The value *ffff* is a 1- to 4-digit decimal number with leading zeros removed. A value over 100% indicates that more work was accomplished with multithreading enabled. If all cores of this type were recently varied online, or if there was an error collecting data, the value will be "----".

### MaxCF *mmmm*%

indicates the multithreading maximum capacity factor for cores of a certain CPU type. The multithreading maximum capacity factor is an estimate of the ratio of the maximum amount of work for the current workload that could be accomplished if all activated threads of the core were kept running, to the amount of work that would have been accomplished if multithreading was disabled. The value *mmmm* is a 1- to 4-digit decimal number with leading zeros removed. If every thread can run this workload concurrently with no interference from the activity of other threads on the same core, the MaxCF value is the number of threads multiplied by 100%. If all cores of this type were recently varied online, or if there was an error collecting data, the value will be "----".

Response 5:

```
A speed change was detected during this interval.
```

is displayed if a change in the speed for a processor type was detected during this interval. In this case, the statistics for utilization, productivity, capacity factor, and maximum capacity factor might not be valid. Wait until the next interval to get a valid sample. This line will appear after the capacity factor information for each core type affected by the change.

Response 6:

```
A multithreading configuration change was detected during this
interval for CPU types <CP > <ZIIP> <ICF > <IFL >
```

is displayed if a change in the multithreading configuration for a processor type occurred during the interval or is in progress due to a request made by a SET MULTITHREAD command. This response line follows the rest of the normal response and includes a list of the CPU types affected by the change. In the response for each of the affected CPU types, the statistics for utilization, productivity, capacity factor, and maximum capacity factor will usually contain dashes if they cannot be calculated due to the transition. Wait until the next interval to get a valid sample.

## Example

If the ENABLE operand was specified on the MULTITHREADING statement in the system configuration file, and you issue the INDICATE MULTITHREAD command on a system that is capable of multithreading, you might see a response similar to the following example.

```
indicate mt
Multithreading is enabled.
Statistics from the interval 13:00:30 - 13:01:00
Core Type CP   Busy  8%   TD  1.00 of 1   Prod 100%   Util  8%
   CF 100%   MaxCF 100%
Core Type IFL  Busy  67%  TD  1.24 of 2   Prod  82%   Util  55%
   CF 107%   MaxCF 131%
Core Type ZIIP Busy  5%   TD  1.00 of 1   Prod 100%   Util  5%
   CF 100%   MaxCF 100%
```

If a speed change was detected for CP and specialty processors, you might see a response similar to the following example.

```
indicate mt
Multithreading is enabled.
Statistics from the interval 13:00:30 - 13:01:00
Core Type CP   Busy  8%   TD  1.00 of 1   Prod 100%   Util  8%
```

## INDICATE MULTITHREAD

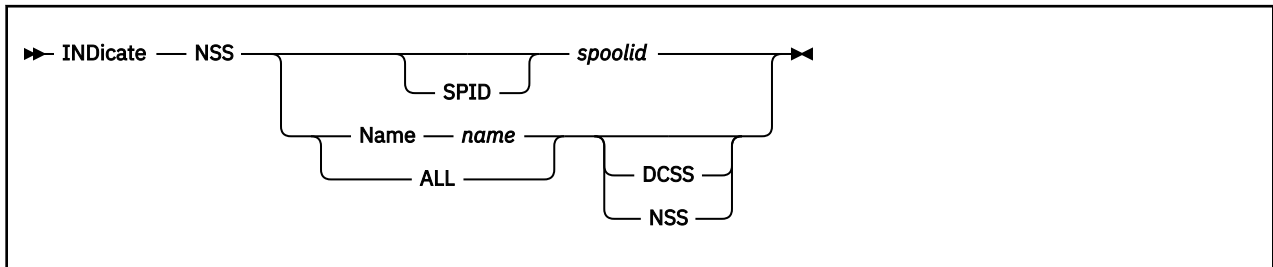
```
CF 100% MaxCF 100%
A speed change was detected during this interval.
Core Type IFL Busy 67% TD 1.24 of 2 Prod 82% Util 55%
CF 107% MaxCF 131%
A speed change was detected during this interval.
Core Type ZIIP Busy 5% TD 1.00 of 1 Prod 100% Util 5%
CF 100% MaxCF 100%
A speed change was detected during this interval.
```

If an SMT configuration change is detected for IFL processors, you might see a response similar to the following example.

```
indicate mt
Multithreading is enabled.
Statistics from the interval 16:07:30 - 16:08:00
Core Type CP Busy 8% TD 1.00 of 1 Prod 100% Util 8%
CF 100% MaxCF 100%
Core Type IFL Busy ---% TD ----- of -- Prod ---% Util ---%
CF ---% MaxCF ---%
Core Type ZIIP Busy 5% TD 1.00 of 1 Prod 100% Util 5%
CF 100% MaxCF 100%
A multithreading configuration change was detected during this
interval for CPU types IFL
```

For an example showing the relationship among the INDICATE MULTITHREAD, QUERY MULTITHREAD, and QUERY PROCESSORS commands, see [“QUERY MULTITHREAD”](#) on page 1026.

## INDICATE NSS



### Authorization

Privilege Class: E

### Purpose

Use INDICATE NSS to display information about the named saved systems (NSSs) and saved segments that are loaded in the system and are in use by at least one user.

The information returned by the INDICATE NSS command pertains to the system owned storage for NSSs and saved segments, not to any virtual machine's storage.

### Operands

#### SPID

indicates that the next operand is a spool ID.

#### *spoolid*

is the system spool file identifier for a system data file containing an NSS or saved segment.

#### Name *name*

identifies the name of the NSS or saved segment that information is requested about.

#### ALL

indicates a request for information about all NSSs or saved segments that are loaded.

#### DCSS

#### NSS

indicates whether information is requested about saved segments (DCSS operand) or named saved systems (NSS operand). If neither operand is specified, information is returned for both.

### Usage Notes

1. Processing of an INDICATE NSS command might be delayed if DIAGNOSE code X'64' is concurrently performing the initial load of a large DCSS.

### Responses

Response 1:

The system displays this response for each NSS or saved segment.

```
Filename=filename Filetype=filetype Class=cl Spoolid=nnnn
Time loaded=ddd hh:mm Size={nnnnM}
                               {nG}
Pages: Main=nnnnnn Xstore=--- Dasd=nnnnnnnnnn Locked=---
       Instan=n...n Reserved=n...n
Paging:
Xstore: Reads=--- Writes=--- Migrates=---
Dasd: Reads=nnnnnnnnnn Writes=nnnnnnnnnn
```

**Filename=filename**

is the file name of the system data file containing the NSS or saved segment. It is also the name of the NSS or saved segment.

**Filetype=filetype**

is the file type of the system data file containing the NSS or saved segment:

<b>File Type</b>	<b>Meaning</b>
NSS	Named saved system.
DCSS	Discontiguous saved segment (DCSS) below 2047 MB.
DCSSG	DCSS above 2047 MB.
DCSS-S	Segment space.
CPNSS	CP system service NSS.
CPDCSS	CP system service DCSS.

**Class=cl**

is the spool file class of the NSS or saved segment.

**Spoolid=nnnn**

is the system spool file identifier of the system data file containing the NSS or saved segment.

**Time loaded=ddd hh:mm**

is the length of time that this NSS or saved segment has been loaded, in days, hours, and minutes.

**Size=nnnnM****Size=nG**

is the size of the NSS or saved segment (number of 1 MB segments in which pages are defined).

**Pages**

includes information about pages for this NSS or saved segment:

**Main=nnnnnn**

is the number of pages resident in real storage.

**Xstore=---**

is included for compatibility only. It is always displayed as dashes.

**Dasd=nnnnnnnnnn**

is the number of pages allocated on DASD paging space.

**Locked=---**

is included for compatibility only. It is always displayed as dashes.

**Instan=n...n**

is the number of instantiated pages. Since NSS/DCSS pages can exist in multiple levels of the storage hierarchy (for example, central storage and in auxiliary), the instantiated count of pages may be less than the total of the resident and DASD page counts. A copy of a page in multiple places in the paging hierarchy counts as one instantiation.

**Reserved=n...n**

the number of pages allocated for reserved storage.

**Paging**

includes the information about paging for this NSS or saved segment.

**Xstore: Reads=--- Writes=--- Migrates=---**

is included for compatibility only. It is always displayed as dashes.

**Dasd**

includes information about paging between DASD paging space and real storage for this NSS or saved segment.

**Reads=nnnnnnnnnn**

is the total number of page reads.



**Writes=nnnnnnnnnn**

is the total number of page writes.

Response 2:

If ALL was specified and no NSS or saved segment is loaded in the system, the following is displayed:

```
No NSS or DCSS was found
```

Response 3:

If ALL NSS was specified and no NSS is loaded in the system, the following is displayed:

```
No NSS was found
```

Response 4:

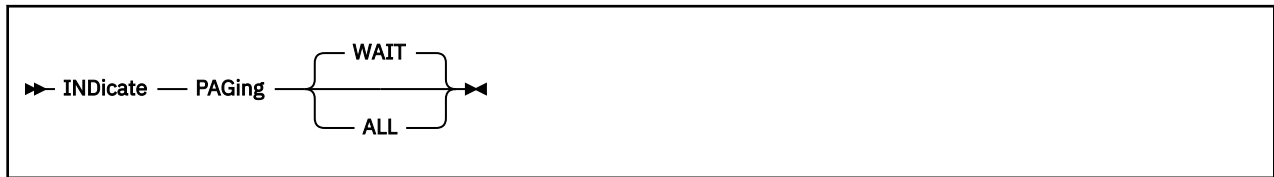
If ALL DCSS was specified and no saved segment is loaded in the system, the following is displayed:

```
No DCSS was found
```

**Messages**

- HCP008E Invalid spoolid-*spoolid*
- HCP1013E An invalid operand was supplied for NAME - *operand*
- HCP1042E NSS or saved segment with {name *name* | spoolid *spoolid*} is not loaded, is a member segment, or does not exist.
- HCP1042E NSS with name *name* is not loaded or does not exist.
- HCP1042E Saved segment with name *name* is not loaded, is a member segment, or does not exist.

## INDICATE PAGING



### Authorization

Privilege Class: E

### Purpose

Use INDICATE PAGING to do the following:

- Display a list of the virtual machines in page wait status
- Display page residency data for all system users

### Operands

#### WAIT

identifies the virtual machines currently in page wait and displays the number of pages allocated on auxiliary storage for each one. This is the default.

Use PAGING WAIT when the INDICATE QUEUES command has shown that a significant proportion of the active users of the system are persistently in page wait.

#### ALL

displays the page residency data for all users of the system.

### Usage Notes

1. The number of pages on DASD will include pages associated with any address space owned by that user.
2. INDICATE PAGING WAIT identifies users in page wait and indicates the number of pages on DASD for that user. A user waiting for a page that is in another user's address space would not be counted in the number of DASD pages reported.
3. If a large virtual machine is logged off multiple times and two or more logged-off instances are still going through memory reset, the user ID (transformed from uppercase to lowercase) will appear multiple times in the responses to this command. Eventually, each instance will finish memory reset and will disappear from the responses.

The following sequence of events is an example of this type of situation:

- a. A user logs on virtual machine "LARGEONE" and runs a workload that instantiates a large amount of memory.
- b. The user logs off the virtual machine. This instance of virtual machine "LARGEONE" becomes known as instance "largeone."
- c. Responses to INDICATE PAGING include "largeone --- yyyyyyyyyy LOGOFF."
- d. The user logs back on to virtual machine "LARGEONE" and runs a workload that instantiates a large amount of memory.
- e. The user logs off. This instance of virtual machine "LARGEONE" also becomes known as instance "largeone."

- f. Responses to INDICATE PAGING now include two occurrences of "largeone --- yyyyyyyyyy LOGOFF."
- g. Eventually instance "largeone" from Step "3.b" on page 546 completes reset and disappears.
- h. Responses to INDICATE PAGING now include only one occurrence of "largeone --- yyyyyyyyyy LOGOFF."
- i. Eventually instance "largeone" from Step "3.e" on page 546 completes reset and disappears.

## Responses

Response 1:

```

  USERID      XSTORE      DASD STATUS
  -----
userid1      --- yyyyyyyyyy zzzzzz
userid2      --- yyyyyyyyyy zzzzzz
userid3      --- yyyyyyyyyy zzzzzz
...
useridn ...

```

### **userid**

identifies the virtual machine.

---

is included for compatibility only. It is always displayed as dashes.

### **yyyyyyyyyy**

is the decimal number of pages this user has on DASD.

### **zzzzzz**

is the virtual machine status and is either LOGOFF, if the user has logged off but its memory is still in the process of being released, or --- otherwise.

Response 2:

```
NO USERS IN PAGE WAIT
```

is issued when appropriate.

## Examples

Example 1:

You can receive responses for one or more virtual machines. For example, the following response indicates that two users are in page wait:

```

indicate paging wait
  USERID      XSTORE      DASD STATUS
  -----
userid1      ---      110 ---
userid2      ---      127 ---

```

## INDICATE QUEUES



### Authorization

Privilege Class: E

### Purpose

Use INDICATE QUEUES to display, in order of their priority, current members of the dispatch and eligible lists.

**Note:** Each **member** is a virtual processor with a transaction in progress. Status critical to the system scheduler and dispatcher is also displayed.

### Operands

#### EXP

displays expanded information about virtual processors with a transaction in progress.

### Usage Notes

1. When INDICATE QUEUES consistently shows *En* virtual processors, an eligible list is forming because one or more system resources appears to be constrained. Under such conditions, forcing users with a transaction in progress to wait in the eligible list increases virtual machine throughput as a whole, and also reduces system overhead. The appearance of a constrained resource is affected by:
  - a. The amount of resources available on the system.
  - b. The total demand for the resources.
  - c. The fraction of resource that is available to certain user classes. For example, the SET SRM STORBUF and SET SRM DSPBUF commands can affect the amount of resources available to certain user classes.

The system resources tracked are: storage, paging capacity, and CPU.

2. When you enter INDICATE QUEUES EXP and the *bbb* field is TI (virtual processor in test-idle state), the associated priority field shows 99999. This should not be read literally as a 99,999 second deadline priority; the 99999 value is used by the scheduler to temporarily place the virtual processor at the bottom of the dispatch list.
3. The format of Response 2 is extremely sensitive to the product release and service level.
4. All virtual processors on the limit list are also part of the dispatch list. However, they have exceeded the limit of the CPU resource that they are allowed to use and will not run until they leave the limit list.
5. CP's virtual processor management has been improved so that no user stays in the eligible list more than an instant before being added to the dispatch list. Therefore users are rarely shown to be in the eligible list.

### Responses

**Notes:** The response you receive lists users in the following order:

1. All dispatch list users (*Qn*) in dispatch priority order
2. All eligible list users (*En*) in eligible list priority order.

The data displayed by this command represents the system at a single, self-consistent instant. (The system is locked briefly so that no changes occur while the data is being gathered.)

Response 1:

```
userid1 ttnn aa bbb s...s/t...t
(1 user per line)
```

**userid1**

**userid2**

each identifies a virtual machine.

**tt**

is a code for the VMATYPE, if other than the user. It can be:

- MP for virtual MP. Each virtual CPU defined for a user shows up separately with the same user ID and a type of MP. Each is a separately dispatchable unit.
- Blank if the virtual processor is any other type.

**nn**

is the virtual CPU number of this virtual processor for virtual MP guests, or is blank if not for virtual MP guests.

**aa**

is the current list status of the user:

**Status**

**List the User Is On**

**Q0**

The dispatch list and exempt from eligible list delays

**Q1**

The dispatch list and entered as an E1 user

**Q2**

The dispatch list and entered as an E2 user

**Q3**

The dispatch list and entered as an E3 user

**E1**

The E1 eligible list

**E2**

The E2 eligible list

**E3**

The E3 eligible list

**LN**

The limit list (subset of dispatch list) due to SHARE LIMITHARD setting.

**L0**

The limit list and exempt from eligible list delays.

**L1**

The limit list and the user entered the dispatch list as a class "1" user.

**L2**

The limit list and the user entered the dispatch list as a class "2" user.

**L3**

The limit list and the user entered the dispatch list as a class "3" user.

**Note:** Users who have been defined as QUICKDSP are in Q0 status whenever they are in the dispatch list. Other users may show up briefly as Q0 if they really belong in the eligible list, but must make brief appearances in the dispatch list (for example, to finish with and release a lock). These users could also appear in the L0 status.

## INDICATE QUEUES

### **bbb**

is one of the following status indicators:

#### **Rnn**

indicates that the user is the current RUNUSER on the specified real processor, where *nn* is the processor ID (in hexadecimal).

#### **PG**

indicates that the user is in page wait; the user's virtual machine is not executing because CP is trying to bring in a page from a paging device.

#### **IO**

indicates that the user is in I/O wait. The user's virtual machine cannot run and has I/O either queued by CP or already in progress.

**Note:** When a user performs I/O through a synchronous DIAGNOSE instruction (as when running CMS), *userid* shows up as I/O, not as EX, while the instruction waits for the I/O.

#### **AP**

indicates that the user is waiting for an APPC/VM function to complete. The APPC/VM function was invoked with the WAIT=YES option.

#### **EX**

indicates that the user is in instruction simulation wait; that is, the user's virtual machine is waiting for the completion of instruction simulation.

#### **PS**

indicates that the user is in PSW wait; that is, the user's virtual machine is in an enabled wait state.

#### **TI**

indicates that a user is in the test-idle state; that is, the user is idle, but has not been idle long enough to be dropped from the dispatch list.

#### **R**

indicates that a user is in the ready state.

–

indicates that a user is in another state not described by the above indicators.

**Note:** In cases where a virtual machine may be in more than one of the above states, only one state is displayed; the state displayed is the first one encountered in the order of priority indicated above.

### **s...s**

is a decimal number indicating the number of pages resident in real storage for this user. The *s...s* value is displayed as 8 - 10 digits.

### **t...t**

is a decimal number indicating the number of pages estimated by the dispatcher as the working set of this user. The *t...t* value is displayed as 8 - 10 digits.

Response 2:

```
userid tnn aa bbb s...s/t...t wxyz sppppp  
(1 user per line)
```

is the response received when INDICATE QUEUES is issued with the EXP option. All fields to the left of *wxyz* are the same as described for Response 1.

### **wxyz**

contains the following 4 status indicators:

#### **w**

##### **H**

The user is receiving a hotshot.

##### **L**

The user is receiving a lock-shot.

**x****I**

The user is receiving interactive bias.

**y**

is one of the following status indicators that refers to a user's eligible list stay, or last eligible list stay if the user is currently in the dispatch list. If more than one of these conditions applies, the one closer to the top of this list applies.

**D**

The user is currently past its eligible list deadline.

**M**

The user was put in the eligible list for exceeding the maximum allowed working set size (WSS).

**P**

The user returned to the eligible list temporarily because of preemption.

**G**

The user returned to the eligible list temporarily for exceeding WSS-growth-limit.

**L**

The user is in the eligible list after having been in the dispatch list for lock-shot.

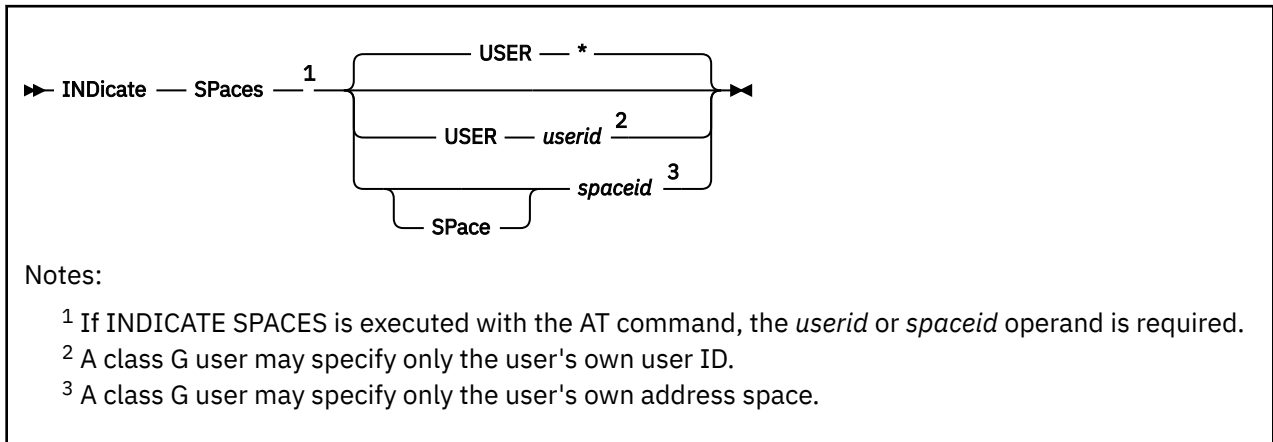
**z****L**

The user is a loading user; that is, the user is considered to be actively bringing in its working set from DASD.

***sppppp***

indicates a user's priority in the eligible or dispatch list, depending on which list the user is in. The priority is expressed in seconds and is represented as an offset from the deadline. The deadline is the time that a dispatch list user is expected to finish its next dispatching minor time slice, or the time that an eligible list user is expected to move to the dispatch list. If the value is zero, the user is at the deadline. If the value is less than zero, the user already has passed the deadline. In this case, the priority is expressed as a negative value and a dash (–) is displayed in the "s" column. If the value is greater than zero, the user has not reached the deadline. The priority is expressed as a positive number, and no sign is displayed in the "s" column.

## INDICATE SPACES



### Authorization

Privilege Class: E, G

### Purpose

Use INDICATE SPACES to display information about a specific address space or all of the address spaces owned by a user. A class G user can display information about the user's own address spaces. A class E user can display information about any address spaces.

### Operands

#### USER \*

returns information about all address spaces owned by the command issuer. This is the default.

#### USER *userid*

returns information about all address spaces owned by *userid*.

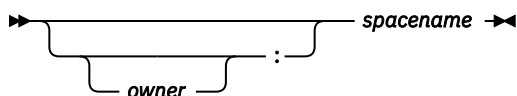
#### SPACE

indicates that the next operand is a *spaceid*.

#### *spaceid*

is the name of the specific space for which data is requested.

The *spaceid* is a 1- to 33-character string of the form:



where *owner* is an optional 1- to 8-character user ID of the virtual machine that owns the address space, and *spacename* is the 1- to 24-character name assigned to the address space. If *owner* is not specified, the command issuer's virtual machine is assumed to be the owner of the address space.

The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

An example of a valid *spaceid* is "OPERATOR:BASE".

Examples of *spaceids* that are not valid are ":BASE" and "BASE". These *spaceids* are not valid only when the command is issued with the AT command.



## Usage Notes

- Any field containing cumulative data from the time the user logged on (reads, writes, and migrates) is preserved during a guest relocation. As a result, the counters might wrap over time.

## Responses

Response 1:

```
Spaceid=ssssssssssssssssssssssssssssssssssssssss Owned size=nu {PRIVATE}
                                                                {SHARED }
Pages: Main=n...n Xstore=--- Dasd=nnnnnnnnnn Locked=---
      Instan=n...n
Private paging:
  Xstore: Reads=---          Writes=---          Migrates=---
  Dasd:   Reads=nnnnnnnnnn   Writes=nnnnnnnnnn
Shared paging:
  Xstore: Reads=---          Writes=---          Migrates=---
  Dasd:   Reads=nnnnnnnnnn   Writes=nnnnnnnnnn
```

### **Spaceid=ssssssssssssssssssssssssssssssssss**

is the fully-qualified address space ID (1 to 33 characters), to which the following data applies.

#### **Owned size=nu**

is the size of the storage within the address space that is owned by the owner of the address space, where  $n$  is a decimal number and  $u$  is the unit of measure. This value can differ from the defined size of the address space only for a primary address space. This value is displayed in units of measure appropriate to the storage size. Shared portions of NSSs or DCSSs that are imbedded in the address space(s) are not considered to be owned by the virtual machine and are not included in the storage size.

<i>Table 20. Storage Units of Measure</i>	
<b>Units suffix</b>	<b>Maximum allowed</b>
K - Kilobytes	2096128
M - Megabytes	17592186044416
G - Gigabytes	17179869184
T - Terabytes	16777216
P - Petabytes	16384
E - Exabytes	16

#### **PRIVATE**

#### **SHARED**

indicates whether the address space is in the private state or in the shareable state.

#### **Pages**

includes information about pages for this space.

#### **Main=n...n**

is the number of pages that are resident in real storage for this address space.

#### **Xstore=---**

is included for compatibility only. It is always displayed as dashes.

#### **Dasd=nnnnnnnnnn**

is the number of pages allocated on DASD paging space for this address space.

#### **Locked=---**

is included for compatibility only. It is always displayed as dashes.

### **Instan=*n...n***

indicates the number of instantiated pages for this address space. A page is considered instantiated when it takes up resource in the paging hierarchy (that is, occupies a real frame or a slot on paging DASD). A copy of a page in multiple places in the paging hierarchy counts as one instantiation.

### **Private paging**

includes information about private paging for this space. The values include counts only for private pages owned by the address space owner. These counts are cumulative for the life of the address space and do not change when the state (private or shared) of the address space changes.

#### **Xstore: Reads=--- Writes=--- Migrates=---**

is included for compatibility only. It is always displayed as dashes.

### **Dasd**

includes information about paging between DASD paging space and real storage.

#### **Reads=*nnnnnnnnnn***

is the number of pages read from DASD to real storage for this address space.

#### **Writes=*nnnnnnnnnn***

is the number of pages written from main storage to DASD for this address space.

### **Shared paging**

includes information about shared paging for this space. The values include counts only for shared pages owned by the address space owner. These counts are cumulative for the life of the address space and do not change when the state (private or shared) of the address space changes.

#### **Xstore: Reads=--- Writes=--- Migrates=---**

is included for compatibility only. It is always displayed as dashes.

### **Dasd**

includes information about paging between DASD paging space and real storage.

#### **Reads=*nnnnnnnnnn***

is the number of pages read from DASD to real storage for this address space.

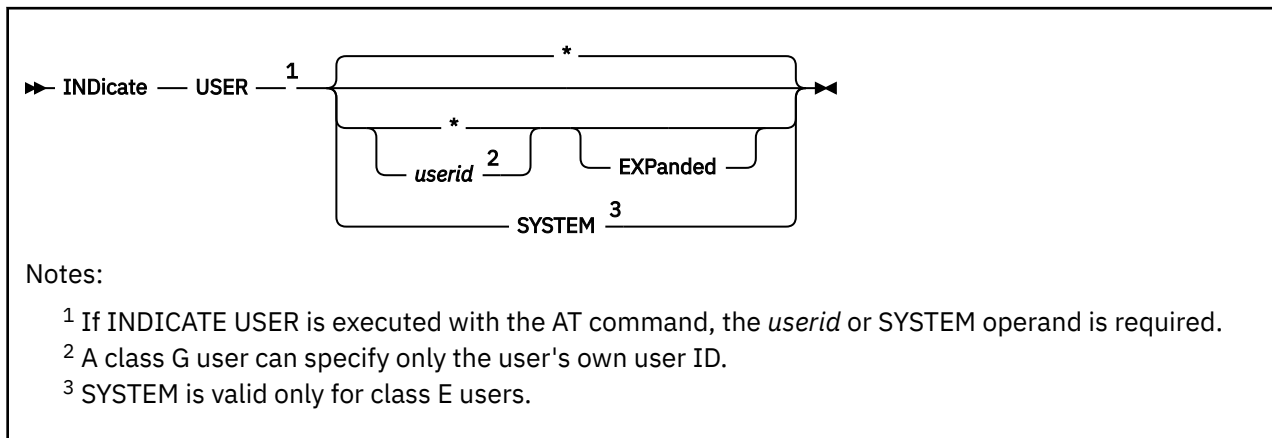
#### **Writes=*nnnnnnnnnn***

is the number of pages written from real storage to DASD for this address space.

## **Messages**

- HCP007E Invalid userid - *userid*.
- HCP1041E Address space *spaceid*. does not exist
- HCP6578E Invalid address space identifier - *option*.

## INDICATE USER



### Authorization

Privilege Class: E, G

### Purpose

Use INDICATE USER to display the resources that are used or occupied by a virtual machine or by the system. A class G user can display information about the user's own virtual machine. A class E user can display information about any virtual machine or the system.

### Operands

\*

#### *userid*

specifies the virtual machine for which you are requesting information. The default is an asterisk (\*), which obtains the information for your own virtual machine.

#### EXPanded


requests an expanded response.

#### SYSTEM

requests information about system resource usage.

See Usage Note “11” on page 556 for a list of the response fields that are meaningful for INDICATE USER SYSTEM.

### Usage Notes

1.  **Attention:** In Response 1, a user's counts for the IO, RDR, PRT, and PCH fields are reset to zero whenever an ACNT command is processed for that user ID. In Response 2 (EXPANDED operand), these fields are not reset when an ACNT command is processed for the user ID.
2. In Response 1, a user's counts for RDR, PRT, and PCH are no longer updated after they reach a value of 999999. These counts are reset to zero when the ACNT command is processed for the user or when the user logs off. If you want to see these counts when they become greater than 999999, refer to Response 2 (EXPANDED operand).
3. Any field containing cumulative data from the time the user logged on (reads, writes, and migrates) is preserved during a guest relocation. As a result, the counters might wrap over time.
4. Only one or no virtual CPUs have an IPLed device or a named saved system that is displayed at any given time. The rest is NONE.

## INDICATE USER

5. Use INDICATE USER to display the total amount of certain resources used by your virtual machine during the current session.
6. Use INDICATE USER before and after the execution of a program to determine the execution characteristics of that program in terms of the resources used.
7. All displayed numbers are in decimal, except for *vdev* and CPU *nn*.
8. When the wait-state interpretation capability is used, the guest virtual machine remains dispatched, even when it enters a PSW enabled-wait state. Time spent dispatched in the wait state is included in the virtual and total CPU time reported for the guest.
9. For compatibility, CP continues to accept the designation XA. XA and ESA are synonymous.
10. If the EXPANDED response is given for a user who is logging on or off and does not have a base address space, then the primary space name and primary space type will be filled with asterisks in the response.
11. The following response fields are meaningful for the INDICATE USER SYSTEM response:
  - USERID is always SYSTEM.
  - STOR is the size of that portion of the system execution space below 2G.
  - DEVNUM is the number of CPACCESSed minidisks.
  - Paging statistics (RES, LOCKEDREAL, NPREF, PREF, READS and WRITES) pertain to pages in system-owned CP utility address spaces. Examples are the "system virtual space" that is used for spooling and spaces for virtual free storage, pageable page tables, VDISKS, and ISFC communication.
  - CTIME is time since system IPL or restart.

Not meaningful in INDICATE USER SYSTEM output are: INSTAN, MACH, VIRT, IPLSYS, WS, RESVD, CPU, VTIME, TTIME, IO, RDR, PRT, and PCH.

## Responses

### Response 1:

This is the response format when the EXPANDED operand is *not* specified. This format is also displayed when the SYSTEM operand is specified. See Usage Note “11” on page 556 for more information about what is meaningful and not meaningful in INDICATE USER SYSTEM output.

This response gives all data from the user VMDBK relevant to the user's virtual machine paging activity, resource occupancy, processor usage, and accumulated I/O activity counts since logon. Time and count values are ever-increasing accumulators.

```
USERID=userid MACH={ESA} STOR=nu VIRT=V XSTORE=---
                {XA }
                {XC }
                {Z  }
IPLSYS={DEV vdev} DEVNUM=nnnnn
        {DEV NONE}
        {sysname }
PAGES: RES=n...n WS=n...n LOCKEDREAL=n...n RESVD=n...n
        INSTAN=n...n
NPREF=n...n PREF=0 READS=n...n WRITES=n...n
CPU nn: CTIME=hh:mm VTIME=mmm:ss TTIME=mmm:ss IO=nnnnnn
        RDR=nnnnnn PRT=nnnnnn PCH=nnnnnn TYPE={CP } CPUAFFIN={ON }
                {IFL } {SUPP}
                {ZIIP} {OFF }
                {ICF }
```

### **USERID=*userid***

identifies the virtual machine. If the SYSTEM operand was specified, *userid* is SYSTEM.

**MACH=ESA****MACH=XA****MACH=XC****MACH=Z**

indicates the virtual machine operating mode.

**STOR=*nu***

indicates the virtual machine storage size, where *n* is a decimal number and *u* is the unit of measure. If the SYSTEM operand was specified, this indicates the size of the system execution area of the system execution space.

<i>Table 21. Storage Units of Measure</i>	
<b>Unit of measure (<i>u</i>)</b>	<b>Maximum allowed (<i>n</i>)</b>
K - Kilobytes	2096128
M - Megabytes	17592186044416
G - Gigabytes	17179869184
T - Terabytes	16777216
P - Petabytes	16384
E - Exabytes	16

**VIRT=V**

appears for compatibility purposes.

**XSTORE=---**

is included for compatibility only. It is always displayed as dashes.

**IPLSYS=DEV *vdev*****IPLSYS=DEV NONE****IPLSYS=*sysname***

indicates the device number (*vdev*) of the last IPLed device for the virtual configuration, or the name (*sysname*) of the last named saved system to be IPLed. (The named saved system was previously saved through the SAVESYS command.) DEV NONE indicates that no device or named saved system was IPLed.

**DEVNUM=*nnnnn***

indicates the number of devices that are attached to this user.

**PAGES**

includes information about pages for this user.

**RES=*n...n***

indicates the number of pages for this user that are resident in real storage at the time the command is issued.

**WS=*n...n***

indicates the most recent system estimate of the user's working set size.

**LOCKEDREAL=*n...n***

indicates the current number of pages for this user that are locked to real storage frames.

**RESVD=*n...n***

indicates the current number of real storage frames that are reserved for pages for this user.

**INSTAN=*n...n***

indicates the number of instantiated pages for this user. A page is considered instantiated when it takes up resource in the paging hierarchy (that is, occupies a real frame or a slot on paging DASD). A copy of a page in multiple places in the paging hierarchy counts as one instantiation.

## INDICATE USER

### **NPREF=*n...n***

indicates the number of private address space pages owned by this user that are currently residing on paging space. If the SYSTEM operand was specified, this indicates the number of shared address space pages that are owned by the system that are currently residing on paging space.

### **PREF=0**

appears for compatibility purposes.

### **READS=*n...n***

indicates the total number of page reads for this user since logon.

### **WRITES=*n...n***

indicates the total number of page writes for this user since logon.

### **CPU *nn***

indicates to which virtual processor the following information refers.

### **CTIME=*hh:mm***

indicates total connect time for the user, in hours and minutes.

### **VTIME=*mmm:ss***

indicates the total virtual time for the user, in minutes and seconds. If multithreading is enabled, this is reported as MT-1 equivalent time.

### **TTIME=*mmm:ss***

indicates the total virtual and simulation time for the user, in minutes and seconds. If multithreading is enabled, this is reported as MT-1 equivalent time.

### **IO=*nnnnnn***

indicates the total number of nonspooled I/O requests issued by the user since the user logged on, or an ACNT command was issued for the user.

### **RDR=*nnnnnn***

indicates the total number of virtual cards read since the user logged on or an ACNT command was issued for the user.

### **PRT=*nnnnnn***

indicates the total number of virtual lines printed since the user logged on or an ACNT command was issued for the user.

### **PCH=*nnnnnn***

indicates the total number of virtual cards punched since the user logged on or an ACNT command was issued for the user.

### **TYPE=CP**

indicates the virtual CPU is a central processor.

### **TYPE=IFL**

indicates the virtual CPU is an IBM Integrated Facility for Linux. The virtual CPU can run on a real IFL or it can be simulated.

### **TYPE=ZIIP**

indicates the virtual CPU is an IBM z Integrated Information Processor. The virtual CPU can run on a real zIIP or it can be simulated.

### **TYPE=ICF**

indicates the Internal Coupling Facility. The virtual CPU can run on a real ICF or it can be simulated.

### **CPUAFFIN=ON**

indicates CPU affinity has been set on with the SET CPUAFFINITY command and this virtual CPU is dispatched on a real CPU of the same TYPE.

### **CPUAFFIN=SUPP**

indicates CPU affinity is suppressed. CPU affinity has been set on with the SET CPUAFFINITY command but there is currently no matching real processor on which to run.

**CPUAFFIN=OFF**

indicates CPU affinity has not been set on. When OFF is in effect the virtual CPU is dispatched on only real CPUs of primary type.

If your virtual machine is a virtual multiprocessor (MP) configuration, the first response is for the base processor. Responses for the additional virtual processors follow the response for the base.

The following fields are meaningful only for the virtual MP configuration as a whole:

MACH  
STOR  
VIRT  
DEVNUM  
RES  
LOCKEDREAL  
RESVD  
NPREF  
PREF  
READS  
WRITES  
WS

The values in the response for additional virtual processors in your configuration are identical with the values in the base processor response for the fields previously listed.

CTIME represents the total connect time only in the base processor response. For your other virtual processors, CTIME indicates the total elapsed time since creation of the virtual processor. If you use the DEFINE command to define the virtual processor after you logged on, this value differs from the value returned in the base processor response.

The following fields are tracked separately for each virtual processor in your virtual machine configuration:

VTIME  
TTIME  
IO  
RDR  
PRT  
PCH  
TYPE  
AFFINITY

IPLSYS reflects the device number of the last device or the name of the last named saved system IPLed on that virtual configuration.

Response 2:

This is the response format when the EXPANDED operand is specified.

This response gives all data from the user VMDBK relevant to the user's virtual machine paging activity, resource occupancy, processor usage, and accumulated I/O activity counts since logon. Time and count values are ever-increasing accumulators.

This response is repeated once for each virtual multiprocessor (MP) configuration owned by the specified user.

```
Userid=userid Mach={ESA} V=V Attached xstore=---
                {XA }
                {XC }
                {Z }
Iplsys={DEV vdev} Devnum=nnnnn SecureIpl={No }
        {DEV NONE}          {Yes}
        {sysname }
Spool: Reads=nnnnnnnnnn Writes=nnnnnnnnnn
```

## INDICATE USER

```
Owned spaces: Number=nnnnn Owned size=nu
Pages: LockedReal=n...n LockedLogical=n...n
Primary space: ID=spaceid {PRIVATE}
                {SHARED }
Defined size=nu Address limit=nu
Private spaces: Number=nnnnn Owned size=nu
Pages: Main=n...n Xstore=--- Dasd=nnnnnnnnnn
        Instan=n...n
        WS=n...n Reserved=n...n
        ResidentLogical=n...n LockedLogical=n...n
Shared spaces: Number=nnnnn Owned size=nu
Pages: Main=n...n Xstore=--- Dasd=nnnnnnnnnn
        Instan=n...n
        ResidentLogical=n...n LockedLogical=n...n
Private paging:
Xstore: Reads=--- Writes=--- Migrates=---
Dasd: Reads=nnnnnnnnnn Writes=nnnnnnnnnn
Shared paging:
Xstore: Reads=--- Writes=--- Migrates=---
Dasd: Reads=nnnnnnnnnn Writes=nnnnnnnnnn
VCONFIG: MODE={ESA390 }
            {GENERAL}
            {LINUX }
            {VM }
            {CF }
CPU nn: Ctime=ddd hh:mm:ss Vtime=ddd hh:mm:ss Ttime=ddd hh:mm:ss
Rdr=nnnnnnnnnn Prt=nnnnnnnnnn Pch=nnnnnnnnnn IO=nnnnnnnnnn
Type={CP } CPUAffinity={ON }
      {IFL } {SUPP}
      {ZIIP} {OFF }
      {ICF }
VtimePrimary=ddd hh:mm:ss TtimePrimary=ddd hh:mm:ss
VtimeSecondary=ddd hh:mm:ss TtimeSecondary=ddd hh:mm:ss
```

### **Userid=userid**

identifies the virtual machine.

### **Mach=ESA**

### **Mach=XA**

### **Mach=XC**

### **Mach=Z**

indicates the virtual machine operating mode.

### **V=V**

appears for compatibility purposes.

### **Attached xstore=---**

is included for compatibility only. It is always displayed as dashes.

### **Iplsys=DEV vdev**

### **Iplsys=DEV NONE**

### **Iplsys=sysname**

indicates the device number (*vdev*) of the last IPLed device for the virtual configuration, or the name (*sysname*) of the last named saved system to be IPLed. (The named saved system was previously saved through the SAVESYS command.) DEV NONE indicates that no device or named saved system was IPLed.

### **Devnum=nnnnn**

indicates the number of devices attached to this user.

### **SecureIpl=No**

### **SecureIpl=Yes**

indicates whether the last IPL of this virtual machine was a secure IPL. A secure IPL occurs when LOADDEV or DUMPDEV parameters are set and the IPL is started by using the LOADDEV or DUMPDEV operand of the IPL command or IPL directory statement.

### **Spool**

includes information about utilization of DASD spooling space by this user.

### **Reads=nnnnnnnnnn**

indicates the total number of spooling page reads by this user since this user logged on.



**Writes=nnnnnnnnnn**

indicates the total number of spooling page writes by this user since this user logged on.

**Owned spaces**

includes information about all address spaces owned by this virtual machine.

**Number=nnnnn**

indicates the total number of address spaces, shared or private, owned by this virtual machine.

**Owned size=nu**

indicates the total size of all storage owned by this virtual machine in all address spaces, shared or private, owned by this virtual machine, where  $n$  is a decimal number and  $u$  is the unit of measure (see [Table 21 on page 557](#)).

Shared portions of a NSS or DCSS that are imbedded in the address space(s) are not considered to be owned by the virtual machine and thus are not included in the storage size. The size is displayed in units of measure appropriate to the storage size. Owned sizes which are a multiple of kilobytes, but larger than 16 megabytes, are rounded to the nearest megabyte.

**Pages**

includes information about pages for this user.

**LockedReal=n...n**

indicates the current number of pages for this user that are locked to real storage frames.

**LockedLogical=n...n**

indicates the current number of pages for this user that are locked to host logical storage pages.

**Primary space**

includes information about the user's base address space.

**ID=spaceid**

indicates the fully-qualified space ID (3 to 33 characters) of the user's base address space.

**PRIVATE  
SHARED**

indicates whether the user's base space is private or shared.

**Defined size=nu**

indicates the size of the base address space, where  $n$  is a decimal number and  $u$  is the unit of measure (see [Table 21 on page 557](#)).

**Address limit=nu**

indicates the maximum address addressable by the virtual machine in the base address space, plus one, where  $n$  is a decimal number and  $u$  is the unit of measure (see [Table 21 on page 557](#)). This value includes any discontinuous shared storage.

**Private spaces**

includes information about all private address spaces owned by this virtual machine.

**Number=nnnnn**

indicates the total number of private address spaces owned by this virtual machine.

**Owned size=nu**

indicates the total size of all storage owned by this virtual machine in all private address spaces owned by this virtual machine, where  $n$  is a decimal number and  $u$  is the unit of measure (see [Table 21 on page 557](#)).

Shared portions of NSSs or DCSSs that are imbedded in the address space(s) are not considered to be owned by the virtual machine and thus are not included in the storage size.

**Pages**

includes information about user-owned pages for all private spaces owned by this user.

**Main=n...n**

indicates the number of pages that are resident in real storage for all private spaces owned by this user.

**Xstore=---**

is included for compatibility only. It is always displayed as dashes.

**Dasd=nnnnnnnnnn**

indicates the number of pages allocated on system paging space for all private spaces owned by this user.

**Instan=n...n**

indicates the number of instantiated pages for all private spaces owned by this user. A page is considered instantiated when it takes up resource in the paging hierarchy (that is, occupies a real frame or a slot on paging DASD). A copy of a page in multiple places in the paging hierarchy counts as one instantiation.

**WS=nnnnnn**

indicates the most recent system estimate of the user's working set size. This working set size only accounts for all private spaces owned by this user.

**Reserved=nnnnnn**

indicates the number of real storage frames that are reserved for pages for all private spaces owned by this user.

**ResidentLogical=n...n**

indicates the number of pages that are resident in host logical storage for all private spaces owned by this user.

**LockedLogical=n...n**

indicates the number of pages that are locked to host logical storage for all private spaces owned by this user.

**Shared spaces**

includes information about all shared address spaces owned by this virtual machine.

**Number=nnnnn**

indicates the total number of shared address spaces owned by this virtual machine.

**Owned size=nu**

indicates the total size of all storage owned by this virtual machine in all shared address spaces owned by this virtual machine, where  $n$  is a decimal number and  $u$  is the unit of measure (see [Table 21 on page 557](#)).

Shared portions of NSSs or DCSSs that are imbedded in the address space(s) are not considered to be owned by the virtual machine and thus are not included in the storage size. Owned sizes which are a multiple kilobytes but larger than 16 megabytes, are rounded to the nearest megabyte.

**Pages**

includes information about user-owned pages for all shared spaces owned by this user.

**Main=n...n**

indicates the number of pages that are resident in real storage for all shared spaces owned by this user.

**Xstore=---**

is included for compatibility only. It is always displayed as dashes.

**Dasd=nnnnnnnnnn**

indicates the number of pages that are allocated on system paging space for all shared spaces owned by this user.

**Instan=n...n**

indicates the number of instantiated pages for all shared spaces owned by this user. A page is considered instantiated when it takes up resource in the paging hierarchy (that is, occupies a real frame or a slot on paging DASD). A copy of a page in multiple places in the paging hierarchy counts as one instantiation.

**ResidentLogical=n...n**

indicates the number of pages that are resident in host logical storage for all shared spaces owned by this user.

**LockedLogical=n...n**

indicates the number of pages that are locked to host logical storage for all shared spaces owned by this user.

**Private paging**

includes information about paging of user-owned private pages for this virtual machine.

**Xstore: Reads=--- Writes=--- Migrates=---**

is included for compatibility only. It is always displayed as dashes.

**Dasd**

includes information about DASD paging for all private spaces owned by this user.

**Reads=nnnnnnnnnn**

indicates the total number of user-owned pages read from DASD paging space to real storage for all private spaces owned by this user since the user logged on. This number reflects private reads for all spaces owned by this user, regardless of whether the space is currently private or shared. This count does not include shared pages read to a private space.

**Writes=nnnnnnnnnn**

indicates the total number of user-owned pages written to DASD paging space from real storage for all private spaces owned by this user since the user logged on. This number reflects private writes for all spaces owned by this user, regardless of whether the space is currently private or shared. This count does not include shared pages written from a private space.

**Shared paging**

includes information about paging of user-owned shared pages for this virtual machine.

**Xstore: Reads=--- Writes=--- Migrates=---**

is included for compatibility only. It is always displayed as dashes.

**Dasd**

includes information about DASD paging for all shared spaces owned by this user.

**Reads=nnnnnnnnnn**

indicates the total number of user-owned pages read from DASD paging space to real storage for all shared spaces owned by this user since the user logged on. This number reflects reads of user-owned shared pages for all spaces owned by this user, regardless of whether the space is currently private or shared.

**Writes=nnnnnnnnnn**

indicates the total number of user-owned pages written to DASD paging space from real storage for all shared spaces owned by this user since the user logged on. This number reflects writes of user-owned shared pages for all spaces owned by this user, regardless of whether the space is currently private or shared.

**Note:** The section beginning with CPU *nn* includes information about each virtual CPU in the virtual machine configuration. This section is repeated for each virtual CPU in the virtual machine configuration.

**VCONFIG: MODE=ESA390**

indicates the virtual configuration mode of the virtual machine is ESA390.

**VCONFIG: MODE=GENERAL**

indicates the virtual configuration mode of the virtual machine is GENERAL. VCONFIG modes ESA390 and GENERAL are functionally equivalent. ESA390 is displayed when MACHINE is set to XA, ESA, or XC, and GENERAL is displayed when MACHINE is set to Z.

**VCONFIG: MODE=LINUX**

indicates the virtual configuration mode of the virtual machine is LINUX.

**VCONFIG: MODE=VM**

indicates the virtual configuration mode of the virtual machine is VM.

**VCONFIG: MODE=CF**

indicates the virtual configuration mode of the virtual machine is CF.

**CPU nn**

indicates the virtual processor to which the following information refers.

**Ctime=ddd hh:mm:ss**

indicates the total create time for the virtual CPU in days, hour, minutes, and seconds.

**Vtime=ddd hh:mm:ss**

indicates the total virtual time for the virtual CPU in days, hours, minutes, and seconds. If multithreading is enabled, this is reported as MT-1 equivalent time.

**Ttime=ddd hh:mm:ss**

indicates the total virtual and simulation time for the virtual CPU in days, hours, minutes, and seconds. If multithreading is enabled, this is reported as MT-1 equivalent time.

**Rdr=nnnnnnnnnn**

indicates the total number of virtual cards read by this virtual CPU since the virtual CPU was created.

**Prt=nnnnnnnnnn**

indicates the total number of virtual lines printed by this virtual CPU since the virtual CPU was created.

**Pch=nnnnnnnnnn**

indicates the total number of virtual cards punched by this virtual CPU since the virtual CPU was created.

**IO=nnnnnnnnnn**

indicates the total number of nonspooled I/O requests issued by this virtual CPU since the virtual CPU was created.

**Type=CP**

indicates that the virtual CPU is a central processor.

**Type=IFL**

indicates that the virtual CPU is an IBM Integrated Facility for Linux. The virtual CPU can run on a real IFL or it can be simulated.

**Type=ZIIP**

indicates that the virtual CPU is an IBM z Integrated Information Processor. The virtual CPU can run on a real zIIP or it can be simulated.

**Type=ICF**

indicates that the virtual CPU is an Internal Coupling Facility. The virtual CPU can run on a real ICF or it can be simulated.

**CPUAffinity=ON**

indicates that CPU affinity has been set on with the SET CPUAFFINITY command and this virtual CPU is dispatched on a real CPU of the same TYPE.

**CPUAffinity=SUPP**

indicates that CPU affinity is suppressed. CPU affinity has been set on with the SET CPUAFFINITY command but there is currently no matching real processor on which to run.

**CPUAffinity=OFF**

indicates that CPU affinity is not set on. When OFF is in effect the virtual CPU is dispatched on only real CPUs of primary type.

**VtimePrimary=ddd hh:mm:ss**

indicates for the virtual CPU the total virtual time consumed while running on a primary processor. The time is displayed in days, hours, minutes, and seconds. If multithreading is enabled, this is reported as MT-1 equivalent time.

**TtimePrimary=ddd hh:mm:ss**

indicates for the virtual CPU the total virtual and simulation time while running on a primary processor. The time is displayed in days, hours, minutes, and seconds. If multithreading is enabled, this is reported as MT-1 equivalent time.

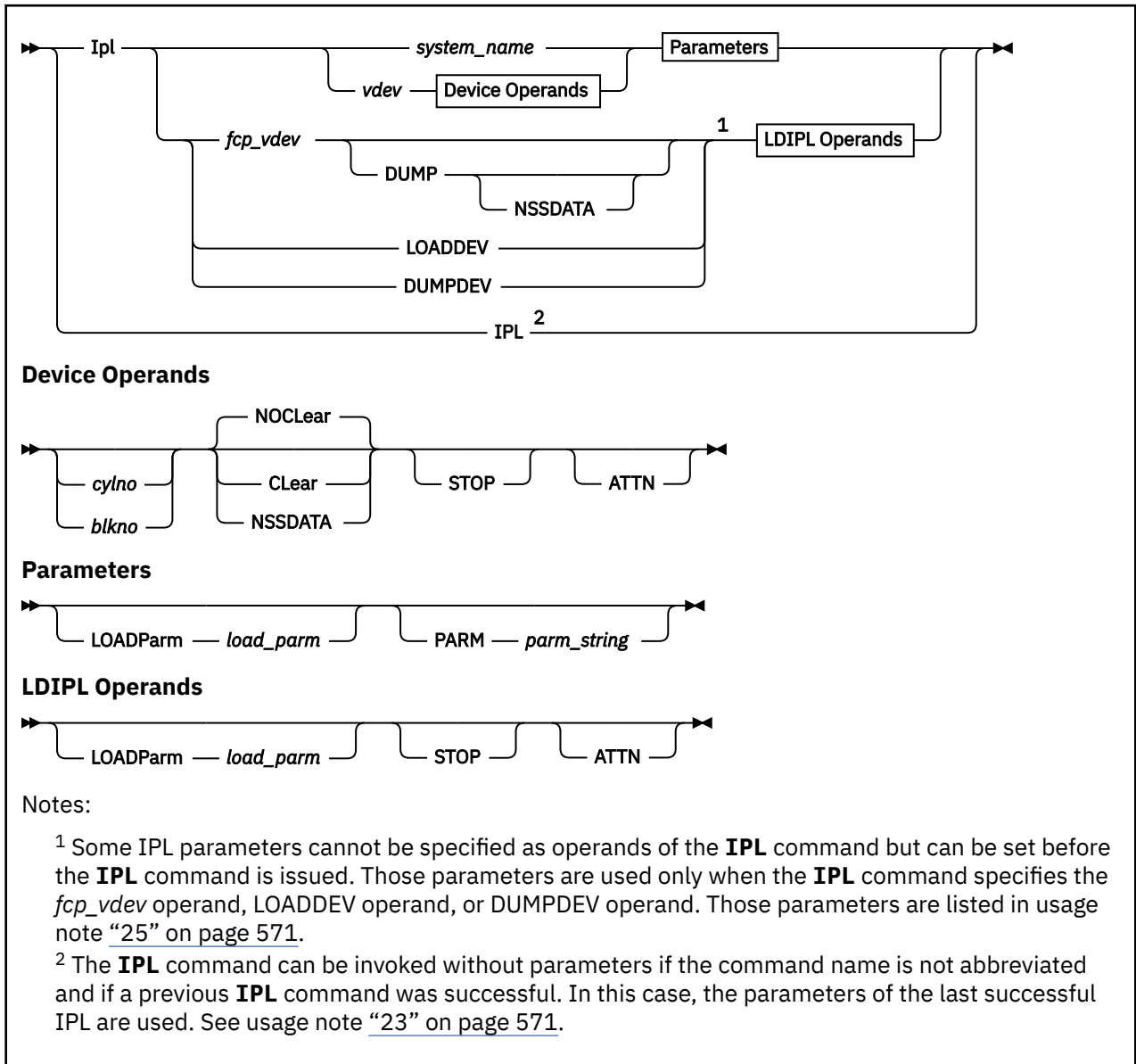
**VtimeSecondary=ddd hh:mm:ss**

indicates for the virtual CPU the total virtual time consumed while running on a secondary processor. The time is displayed in days, hours, minutes, and seconds. If multithreading is enabled, this is reported as MT-1 equivalent time.

**TtimeSecondary=ddd hh:mm:ss**

indicates for the virtual CPU the total virtual and simulation time while running on a secondary processor. The time is displayed in days, hours, minutes, and seconds. If multithreading is enabled, this is reported as MT-1 equivalent time.

## IPL



## Authorization

Privilege Class: G

## Purpose

Use the **IPL** command to simulate an initial program load (IPL) function for your virtual machine.

The **IPL** command can specify four methods to IPL a virtual machine:

- Load a named saved system.
- Use an ECKD device that is specified by the **IPL** command. Optionally specify ECKD device parameters.
- Use a SCSI device that is specified by the **IPL** command's *fcp\_vdev* operand and use required IPL information that is specified in **LOADDEV** or **DUMPDEV** parameters. A list-directed IPL is initiated.

- Use required IPL information, including the virtual device number, that is specified in LOADDEV or DUMPDEV parameters. The device can be an ECKD device or a SCSI device. A list-directed IPL is initiated.

The **IPL** command can specify optional IPL information for any of the four methods.

#### Notes:

1. The **IPL** command is not supported for secondary processors.

## Operands

### *system\_name*

The *system\_name* operand specifies the 1- to 8-character alphanumeric name of a named saved system. The system was saved by using the **SAVESYS** command.

### *vdev*

The *vdev* operand specifies the virtual device number of a device to IPL. The value must be a hexadecimal number in the range X'0000' - X'FFFF'. A guest IPL from the specified device is initiated.

**Note:** DASD that is defined as being unsupported cannot be IPLed by CP. Unsupported DASD can be used only by a virtual machine that is IPLed from a supported device or from a named saved system. See [Direct access storage devices \(DASD\) in z/VM: General Information](#).

### *fcp\_vdev*

The *fcp\_vdev* operand initiates a list-directed IPL operation from an FCP-attached (SCSI) device. The *fcp\_vdev* operand specifies the virtual device number of the attached FCP device.

If the **IPL** command's DUMP operand is specified, some storage is saved for the dump. The system dumper program is then loaded and stores data to the device that it is configured to use.

The *fcp\_vdev* operand indicates a list-directed IPL, in which some required IPL information is specified by LOADDEV or DUMPDEV parameters. For more information, see usage notes [“25” on page 571](#) and [“26” on page 573](#).

If the LOADDEV or DUMPDEV parameters specify a virtual device number, you can omit the **IPL** command's *fcp\_vdev* operand and instead use the **IPL** command's LOADDEV or DUMPDEV operand. If both the **IPL** command's *fcp\_vdev* operand and the LOADDEV or DUMPDEV parameters specify a virtual device number, then the *fcp\_vdev* operand must specify the same virtual device number as the LOADDEV or DUMPDEV parameters specify.

The *fcp\_vdev* operand is not allowed if the LOADDEV or DUMPDEV parameters specify the SECURE parameter.

If the *fcp\_vdev* and NSSDATA operands are specified, the DUMP operand must also be specified.

## LOADDEV

### DUMPDEV

The LOADDEV and DUMPDEV operands initiate a list-directed IPL operation from an FCP-attached (SCSI) device or an ECKD device. All information that is required for IPL, including the device number, must be set in the LOADDEV or DUMPDEV parameters.

The LOADDEV operand indicates a list-directed IPL that uses preset LOADDEV parameters to load an operating system program without dump.

The DUMPDEV operand indicates a list-directed IPL that uses preset DUMPDEV parameters to load a system dumper program. The device that is IPLed is also used to store the dump data.

A secure IPL is possible only when the SECURE parameter is set and the **IPL** command specifies the LOADDEV or DUMPDEV operand.

For more information, see usage notes [“25” on page 571](#) and [“26” on page 573](#).

***cylno******blkno***

The *cylno* and *blkno* operands specify the location of the IPL data when the *vdev* operand specifies a DASD device. For CKD and ECKD DASD devices, the *cylno* operand specifies the cylinder number that contains the IPL data. For FBA DASD devices, the *blkno* operand specifies the starting block number where the IPL data is located. If not specified and the *vdev* operand identifies a DASD device, virtual cylinder 0 or block number 0 are defaulted. This operand is only for virtual direct access storage devices.

**NOClear**

The NOCLEAR operand allows the contents of your virtual machine's storage to remain unchanged before program load. NOCLEAR is the default.

**Clear**

The CLEAR operand sets the contents of your virtual machine's storage to binary zeros before the operating system is loaded.

**DUMP**

The DUMP operand initiates a guest IPL from SCSI disk with dump. The following requirements apply when the DUMP operand is specified:

- The *fcp\_vdev* operand specifies an attached FCP device.
- DUMPDEV parameters were set by using the **SET DUMPDEV** command. If the DUMPDEV parameters include a device number, then the *fcp\_vdev* operand matches that device number.

A list-directed IPL with dump is similar to a list-directed IPL without dump but has three differences:

1. Before the machine loader is loaded into storage, a model-dependent amount of guest storage is saved in simulated HSA. The anchor and size of the storage area is saved in the base VMDBK of the guest that requests the dump.
2. A system image is not loaded from the SCSI device that is identified by using a **SET LOADDEV** command. A system dumper is loaded from the SCSI device that is identified by the **SET DUMPDEV** command. The same device is also used to store the dump data. The operands that are valid for the **SET LOADDEV** command are the same operands that are valid for the **SET DUMPDEV** command.
3. The system dumper calls DIAG X'308' with subcode 2 to release the data that is saved in simulated HSA.

**NSSDATA**

The NSSDATA operand preserves a private copy of the contents of the existing embedded NSS (named saved system) in memory after the **IPL** command completes.

If the NSSDATA operand is specified when an FCP device is IPLed, then the DUMP operand must also be specified.

Because the NSSDATA operand is not permitted when the LOADDEV or DUMPDEV operand is specified, the NSSDATA operand cannot be used with a secure IPL. The NSSDATA operand is deprecated and might be removed in the future.

The following requirements apply when the DUMP operand is specified:

- The NSS is fully contained within the first extent of guest memory.
- The NSS does not contain SW, SN or SC pages.
- The NSS is not a VMGROUP NSS.

If any of these requirements are not satisfied, the command fails and the system remains unchanged.

**STOP**

The STOP operand halts the virtual machine during the IPL procedure just after the initial PSW is loaded. After the halt, you can simulate an IPL procedure for your virtual machine in instruction step mode. When your virtual machine stops, you can enter CP commands. To restart the virtual machine, enter the **BEGIN** command.



**ATTN**

The ATTN operand sends an attention interrupt to the virtual machine during the IPL procedure.

**LOADPARM** *load\_parm*

The LOADPARM operand specifies a load parameter of 1-8 characters. The parameter value can be retrieved by the guest operating system during its IPL sequence. Its use is determined by the guest operating system, such as indicating the nucleus to be loaded during the IPL sequence. If the LOADPARM operand is not specified, a load parameter of eight EBCDIC blanks is formed.

Use single quotation marks to preserve leading and embedded blanks. If fewer than eight characters are specified, the parameter value is left-justified and padded with blanks.

The parameter value is converted to uppercase and can include only the uppercase alphabetic characters (A-Z), digits 0-9, the period, and the blank.

**PARM** *parm\_string*

The PARM operand specifies a parameter string up to 64 bytes long to pass to the virtual machine in general purpose registers at completion of IPL. The parameter string begins with the first nonblank character that follows the PARM keyword. It consists of all remaining characters that follow on the command line, including trailing blanks. If you specify other optional parameters on the **IPL** command, you must enter the PARM operand last.

If you are IPLing a device, the parameter string is inserted into the virtual machine registers, four bytes per register, starting with register 0. If the string is less than 64 bytes, a single byte of binary zeros is inserted following the string. If you omit PARM, the virtual machine registers are unchanged. If you IPL an NSS that was defined with the PARMREGS=*m-n* option on the **DEFSYS** command, the parameter string is inserted into the virtual machine registers *m* through *n*. Before insertion, the virtual machine registers *m* through *n*, initialized to binary zeros. For more information, refer to the **DEFSYS** command. If you enter a string that is too large to fit in the designated registers, an error message is issued, and the command is not processed. If you specify PARM followed by all blanks or if you omit PARM, the virtual machine registers specified by the PARMREGS option on the **DEFSYS** command contains all zeros.

If you IPL an NSS that was defined with the PARMREGS=NONE option on the **DEFSYS** command, the PARM option yields an error message. The command is not processed.

If you are IPLing an NSS that was defined without the PARMREGS=*m-n* option on the **DEFSYS** command, the parameter string is inserted into the virtual machine registers 0 - 15. The registers are not first initialized to binary zeros. If you specify PARM followed by all blanks or if you omit PARM, the virtual machine registers are unchanged.

**Usage Notes**

1. The **IPL** command can be entered from the virtual machine console as the first or only command on a CP command line. A CP command line is when TERMINAL MODE is CP or CP READ is displayed or the command line begins with #CP. In that case, VM attempts to break out of certain wait states that might exist. Thus the **IPL** command can be entered to cause breakout of certain virtual machine wait states.

The breakout attempt occurs when the **IPL** command is entered, not when the command is processed. Command errors do not prevent the breakout attempt. In some cases, a breakout resets the virtual machine.

2. If you want to use the AUTOOCR, BATCH, NOSPROF, INSTSEG, or SAVESYS parameters of the **IPL** command, you must have PARMREGS=0-15 in the **DEFSYS** command for the CMS named saved system.
3. IPL simulates the LOAD operation on the real system console. The specified virtual device is accessed, and the required input/output operations are performed to retrieve the IPL data. This does not pertain to named saved systems.
4. Optionally, you can stop the IPL procedure after the virtual PSW is loaded. Also, you can pass parameters to the virtual machine's general registers. When the simulated load function is complete,

CP initiates execution of the virtual machine by loading the IPL PSW, which was stored during the simulation process.

5. If you enter `ipl system_name` or `ipl vdev clear`, the **IPL** command performs a system-reset-clear function on your virtual machine as if the **SYSTEM CLEAR** command were processed.

You can enter `ipl vdev noclear` or enter `ipl vdev` and take the NOCLEAR default. In this case, the **IPL** command performs a system-reset-normal function on your virtual machine as if the **SYSTEM RESET** command were processed.

The effects of a system-reset-normal or system-reset-clear function are described in detail in the description for the **SYSTEM RESET** and **SYSTEM CLEAR** commands.

6. Any logical lines that follow the **IPL** command are ignored because execution of the **IPL** command transfers control from the CP environment to the virtual machine environment.
7. The IPL sequence for the program that you want to IPL must use only format 0 CCWs.
8. After you enter the **IPL** command, the pseudo page fault facility is deactivated. To activate (or reactivate) the pseudo page fault facility, you must enter the **SET PAGEX** command with the ON operand. For more information, see usage note [“2” on page 1594](#) in the description of the **SET PAGEX** command.
9. Care must be used when parameters are passed to a named saved system (*system\_name*). Named saved systems might expect certain registers to be initialized when they are given control. Indiscriminate use of the PARM operand might overlay a previously initialized register, causing unpredictable results.
10. The **IPL** command fails if issued when VMRELOCATE is in progress.
11. When *system\_name* is equal to a hexadecimal device number, the device is IPLed if it exists; otherwise, the named saved segment (NSS) is IPLed.
12. The NSS name `"_0___1__"` is reserved for IBM use.
13. The LOADDEV and DUMPDEV keywords are not valid for use as NSS names.
14. Processing of an **IPL** command on an NSS might be delayed if DIAGNOSE code X'64' is concurrently performing the initial load of a large DCSS.
15. The contents of an NSS preserved by processing the NSSDATA parameter includes storage keys that are accurately preserved except for the reference bit, which is unreliable. The change bit is always off for SR and ER pages of an NSS and they cannot be changed by the user. Therefore, the change bit is never on for preserved SR or ER pages.
16. Immediately after an NSS is restored by the **IPL** command, the virtual machine's architecture mode is determined by its MACHINE setting. In some cases, the virtual machine's architecture mode is also determined by the architecture mode when the NSS was saved. The interaction of the MACHINE setting and the architecture mode when the NSS was saved has the following outcomes:

**MACHINE setting: ESA**

**Architecture mode when NSS was saved: ESA/390 or ESA/XC**

If the MACHINE setting is ESA and the NSS was saved while in ESA/390 mode or ESA/XC mode, then the architecture mode is ESA/390.

**Architecture mode when NSS was saved: z/Architecture or z/XC**

If the MACHINE setting is ESA and the NSS was saved while in z/Architecture mode or z/XC mode, then the architecture mode is z/Architecture.

**MACHINE setting: XC**

**Architecture mode when NSS was saved: ESA/390 or ESA/XC**

If the MACHINE setting is XC and the NSS was saved while in ESA/390 mode or ESA/XC mode, then the architecture mode is ESA/XC.

**Architecture mode when NSS was saved: z/Architecture or z/XC**

If the MACHINE setting is XC and the NSS was saved while in z/Architecture mode or z/XC mode, then the architecture mode is z/XC. However, if the virtual machine's relocation domain excludes z/XC capability, then IPL sets the architecture mode to ESA/XC.

**MACHINE setting: Z**

If the MACHINE setting is Z, then the architecture mode is z/Architecture.

If the architecture mode on completion of the **IPL** command differs from the mode at SAVESYS, the PSW is converted to the appropriate format (that is, from a 64-bit ESA/390 or ESA/XC PSW to a 128-bit z/Architecture or z/XC PSW, or vice versa).

17. If you IPL an NSS that was created with the VMGROUP option on the **DEFSYS** command, your virtual machine becomes a member of the virtual machine group that is known by the NSS name. Members of a virtual machine group are permitted to connect to the signal system service and to provide intragroup signaling. Signaling includes awareness messages (signal-in and signal-out) about members that join or leave the group.

If your machine is a member of a virtual machine group, a user with an appropriate privilege class can authorize trace data recording into a system data file for the group. Issue a **TRSAVE** command and specify FOR *sysname*.

18. If you do not want leading blanks or embedded blanks in the LOADPARM operand, you can specify the parameter value as a single blank-delimited token. Single quotation marks preserve leading and embedded blanks.
19. The LOADPARM operand cannot contain a X'15' character.
20. You can specify both the LOADPARM and PARM options when IPLing by either *vdev* or *system\_name*. Both options can be specified on the same command line. However, if you specify the PARM option, it must be the last option on the command.
21. If the tape drive specified is in a Given State, the issuer must be the user ID to whom the tape drive was given, or the command is not performed.
22. This command terminates all Concurrent Copy (CONCOPY) sessions for your virtual machine.
23. The **IPL** command can be invoked without operands if both of the following conditions are true:
- A previous **IPL** command was successful.
  - The **IPL** command is entered without abbreviation (that is, is entered as IPL rather than I or IP).

In such a case, the operands of the last successful **IPL** command are used.

If the **IPL** command is invoked without operands and either of the conditions are false, then an error message is issued.

24. If an external security manager is installed on your system, you might not be authorized to enter this command. For additional information, contact your security administrator.
25. A list-directed IPL is an IPL that is initiated when an **IPL** command or IPL directory statement specifies one of the following operands:
- *fcv\_vdev*
  - LOADDEV
  - DUMPDEV (DUMPDEV is not a valid operand for IPL directory statements)

In these cases, the **IPL** command or IPL directory statement operands do not provide all required information for the IPL. List-directed IPL parameters (LOADDEV and DUMPDEV parameters) provide the IPL information that the **IPL** command or IPL directory statement operands don't provide.

Most list-directed IPL parameters specify information that cannot be entered as operands on the **IPL** command or IPL directory statement. (The one exception is the DEVICE parameter, which value can be specified by the *fcv\_vdev* operand of the **IPL** command or IPL directory statement.)

Table 22 on page 571 lists parameters that are used in list-directed IPLs.

Table 22. LOADDEV and DUMPDEV parameters			
Parameter	Description	Device type	Default value
DEVICE	The number of the virtual device that is IPLed	SCSI and ECKD	There is no default value

Parameter	Description	Device type	Default value
PORTNAME	The fibre channel port name of the device that is IPLed	SCSI only	0 <sup>1</sup>
LUN	The logical unit number of the device that is IPLed	SCSI only	0 <sup>1</sup>
BR_LBA	The logical-block address of the boot record	SCSI only	0
BOOTPROG	The number of the boot program	SCSI and ECKD	0
SECURE/ NOSECURE	Specifies whether the IPL is a secure IPL	SCSI and ECKD	NOSECURE
SCPDATA	Information that is passed to the program that is loaded during IPL	SCSI and ECKD	There is no default value
BOOTREC	The boot record location on the device that is IPLed	ECKD only	LABEL

**Note:**

1. The default value 0 is not a valid device port name and not a valid device LUN. An IPL attempt that uses the default value for the PORTNAME or LUN parameter will fail.

Table 23 on page 572 shows which LOADDEV and DUMPDEV parameters have default values that must be set (changed from the default) before each type of list-directed IPL can be started. Other parameters might need to be set if the default values are not appropriate for your environment.

IPL type	DEVICE parameter	PORTNAME parameter	LUN parameter	SECURE parameter
<b>IPL <i>fcp_vdev</i></b>	The DEVICE parameter is not set or matches <i>fcp_vdev</i> on the <b>IPL</b> command	Required	Required	Not allowed
<b>IPL <i>fcp_vdev</i> DUMP</b>				
<b>IPL LOADDEV</b> from SCSI device	Required	Required	Required	Required if the SECUREIPLREQ option is specified in the OPTION directory statement
<b>IPL LOADDEV</b> from ECKD device	Required	Not applicable	Not applicable	
<b>IPL DUMPDEV</b> from SCSI device	Required	Required	Required	Required if the previous IPL used the SECURE parameter or if the SECUREIPLREQ option is specified in the OPTION directory statement
<b>IPL DUMPDEV</b> from ECKD device	Required	Not applicable	Not applicable	

There are two groups of parameters for list-directed IPLs:

**LOADDEV parameters**

LOADDEV parameters provide IPL specifications when the IPL does not include a dump.

### DUMPDEV parameters

DUMPDEV parameters provide IPL specifications when the IPL does include a dump.

In each group, a value can be set for each parameter that is listed in [Table 22 on page 571](#). The values of the LOADDEV and DUMPDEV parameters are independent of each other and do not affect each other. The settings for LOADDEV parameters are displayed by the **QUERY LOADDEV** command. The settings for DUMPDEV parameters are displayed by the **QUERY DUMPDEV** command.

For more information, see [QUERY LOADDEV](#) and [QUERY DUMPDEV](#) in *z/VM: CP Commands and Utilities Reference*.

26. Parameters for a list-directed IPL without dump (also called LOADDEV parameters) are set by using any of the following methods:

- **SET LOADDEV** command.
- LOADDEV user directory statement. The LOADDEV user directory statement can be modified by a directory management tool such as IBM Directory Maintenance (DirMaint). To set load parameters by using DirMaint, use the **DIRM LOADDEV** command. The LOADDEV user directory statement can be modified also by the `Image_IPL_Characteristics_Define_DM` API function.

Parameters for a list-directed IPL **with** dump (also called DUMPDEV parameters) are set by using the CP **SET DUMPDEV** command.

For more information, see [SET LOADDEV](#) and [SET DUMPDEV](#), [LOADDEV Directory Statement](#) in *z/VM: CP Planning and Administration*, [DIRM LOADDEV](#) in *z/VM: Directory Maintenance Facility Commands Reference*, and [Image\\_IPL\\_Characteristics\\_Define\\_DM](#) API function in *z/VM: Systems Management Application Programming*.

27. A secure IPL is required in the following situations:

- A user's OPTION directory statement includes the SECUREIPLREQUIRED option. For a secure IPL, the **IPL** command must use the LOADDEV operand (for IPL without dump) or the DUMPDEV operand (for IPL with dump).
- A guest's previous IPL was a secure IPL and the current IPL includes a dump. For a secure IPL in this case, the **IPL** command must use the DUMPDEV operand.

28. To initiate a secure IPL by using the **IPL** command, the following conditions must be satisfied:

- The **IPL** command uses the LOADDEV or the DUMPDEV operand.
- The DEVICE parameter specifies the virtual device number.
- The SECURE parameter is set.
- Other list-directed IPL parameters are set to appropriate values for your environment.
- The program that is loaded must be signed. The appropriate certificate must be loaded in the HMC and assigned to the LPAR on which the IPL occurs.

29. z/VM does not support list-directed IPL of a z/VM system from ECKD or a secure IPL of a z/VM system from SCSI or ECKD. The z/VM stand-alone dump utility does not support list-directed IPL from ECKD, and does not support secure IPL from SCSI or ECKD. For more information, see [SET LOADDEV](#) and [SET DUMPDEV](#).

## Options

**Optional CMS Initialization Parameters:** The following CMS parameters are passed to the system profile EXEC (SYSPROF EXEC). Any unrecognized parameters are ignored by CMS initialization, but are still passed to the SYSPROF EXEC.

CMS users can use the PARM operand to specify any of the following CMS parameters:

### AUTOOCR

is the automatic carriage return parameter that simulates the pressing of the ENTER key as input to the virtual machine at the initial VM READ. Your PROFILE EXEC is automatically executed if it exists on your file mode A.

**BATCH**

indicates that the CMS IPL is being performed in a batch instead of an interactive virtual machine.

Use of this parameter does not affect execution of the SYSPROF EXEC. Execution of the SYSPROF EXEC is suppressed only if the NOSPROF parameter was specified.

At the beginning of each job, the batch facility work disk is accessed and immediately erased, preventing the current user job from accessing files that might remain from the previous job. Because of this, execution of the PROFILE EXEC is disabled for the CMS Batch Facility machine. However, if the BATPROF EXEC exists on an accessed system disk, it is invoked instead of the PROFILE EXEC.

**FILEPOOL *filepoolid***

specifies the default file pool that is established as the user's file mode A. If FILEPOOL is not specified, 191 is established as the file mode A.

The following parameters are ignored by the SYSPROF EXEC, but processed by CMS initialization.

**NOSPROF**

indicates that the SYSPROF EXEC is bypassed. You must specify NOSPROF if you intend to enter the **CMSBATCH** command when the initial VM READ is issued. When the SYSPROF EXEC does not exist, you do not have to specify NOSPROF to enter the **CMSBATCH** command.

**INSTSEG YES**

links the default CMS installation saved segment for this CMS session. The CMS installation-saved segment is an optional shared segment that contains EXECs and editor macros that your installation provides.

**INSTSEG NO**

indicates that you do not want to use the CMS installation-saved segment during this CMS session.

**INSTSEG *name***

links the named CMS installation saved segment for this CMS session. The segment can be a physical or a logical saved segment.

**SAVESYS *sysname***

is used while creating named systems. The SAVESYS parameter is available only to Class E users. It saves a virtual machine storage space with registers and PSW as they currently exist. The operand *sysname* must be a predefined name that represents a definition of installation requirements of named system. The definition indicates the number of pages to be saved, the DASD volume on which the system is to be saved, and the shared segments, if any.

After the system is saved, initialization continues, all CMS parameters that are entered on the **IPL** command are ignored, and SYSPROF EXEC is processed. The parameters that are entered on the original **IPL** command are not passed to SYSPROF EXEC.

No other parameters can be specified with this parameter. If any other parameters are specified, all parameters are ignored, and a CONFLICT parameter is passed to SYSPROF EXEC, as well as the parameters entered on the **IPL** command.

**MTSEG YES**

links the default CMS multitasking segment for this CMS session. The CMS multitasking segment is an optional shared segment that contains the CMS multitasking enablement code.

**Note:** The CMS multitasking segment is no longer used because the multitasking enablement code is now included in the CMS nucleus. This parameter has no effect and is retained only for compatibility.

**MTSEG NO**

indicates that you do not want to use the CMS multitasking segment during this CMS session.

**Note:** The CMS multitasking segment is no longer used because the multitasking enablement code is now included in the CMS nucleus. This parameter has no effect and is retained only for compatibility.

**MTSEG *name***

links the named CMS multitasking segment for this CMS session. The segment can be a physical or a logical multitasking segment.

**Note:** The CMS multitasking segment is no longer used because the multitasking enablement code is now included in the CMS nucleus. This parameter has no effect and is retained only for compatibility.

## Optional CMS Initialization usage notes

1. If you want to use the NOSPROF, INSTSEG, or SAVESYS parameters of the **IPL** command, you must have PARMREGS=0-15 specified on the **DEFSYS** command for the CMS named saved system.
2. You can indicate that the CMS initial program load is being performed in a batch rather than an interactive virtual machine. Specify either the BATCH parameter on the **IPL** command or the NOSPROF parameter, and then enter the **CMSBATCH** command at the VM READ. Then, a BATPROF EXEC is processed in the batch machine, rather than a PROFILE EXEC. This option causes the CMS nucleus to issue the **CMSBATCH** command to load the batch processor. Specifying the BATCH option eliminates the need for operator intervention between jobs. See [The CMS Batch Facility in z/VM: CMS Planning and Administration](#) for information.
3. AUTOOCR should not be used for AUTOLOG users if SET AUTOREAD OFF is in the user's profile.

**Note:** For a disconnected CMS service virtual machine, this mechanism can be useful. The name of a CMS restart routine can be specified in the CMS nucleus. In case of an abend, the specified restart routine receives control instead of the service virtual machine entering a disabled wait state. If this routine issues an **IPL CMS PARM AUTOOCR** command, the PROFILE EXEC processes initial setup procedures that allow the service virtual machine to reestablish itself.

4. The SYSPROF EXEC contains some of the CMS initialization function. When you IPL CMS, it is executed by default before any file modes are accessed. The user PROFILE EXEC is invoked after the SYSPROF EXEC. You can bypass the SYSPROF EXEC by specifying the NOSPROF parameter on the **IPL** command.
5. If you defined disks with virtual device numbers 190 and 19E in the system directory, or if they are defined before you IPL CMS, these disks are accessed as the S- and Y-disks. After you IPL CMS, you must enter explicit **ACCESS** commands to access other disks. Ordinarily, you have access only to files with a file mode number of 2 on the system disk.

If you defined disk address 191 and the FILEPOOL parameter is omitted, the 191 disk is accessed as file mode A. However, if the FILEPOOL parameter is specified, your top directory in the identified file pool is accessed as file mode A, even if you also have a 191 minidisk defined. If the access of the top directory fails (perhaps because the file pool is unavailable), the 191 minidisk is accessed. If there is no 191 minidisk, nothing is accessed as file mode A.

When **ACCESS** is the first command that is issued after an IPL of the CMS system, file mode A is not automatically defined. Another **ACCESS** command must be issued to define file mode A.

For more information, see [CMS ACCESS](#) command in *z/VM: CMS Commands and Utilities Reference*.

6. If you have a disk that is defined as virtual device number 192, the following special rules apply when CMS is IPLed:
  - If 192 is an unformatted temporary minidisk or virtual disk in storage, CMS formats it and accesses it as file mode D.
  - If 192 is a CP-formatted temporary minidisk or virtual disk in storage, CMS reformats it for CMS use and accesses it as file mode D.
  - If 192 is a CMS-formatted temporary minidisk, virtual disk in storage, or permanent minidisk that is accessed as a file mode other than D, CMS reaccesses it as file mode D.
  - If 192 is an unformatted or CP-formatted permanent minidisk, CMS does not automatically format, reformat, access, or reaccess it.

When CMS accesses a 192 minidisk as file mode D, any minidisk or SFS directory that is already accessed as D is released.

## Responses

Usually, after a successful IPL, any responses you receive are those from the operating system that was loaded and initialized.

Response 1:

```
Trace active at IPL
```

is issued when trace is active.

Response 2:

```
IPL stop at xxxxxxxx
```

is issued after the IPL is complete when the STOP option is specified.

**XXXXXXXX**

is the address where execution of the loaded program starts.

After this response is issued, your virtual console is placed into CP READ so that you can enter CP commands (for example, **TRACE**). After you enter all wanted CP commands, issue the **BEGIN** command to start the loaded program.

## Messages

**Note:** A guest IPL from a SCSI, NVMe, or ECKD device might cause non-CP messages to be displayed. For descriptions of non-CP messages, see *IBM Z: IPL Machine Loader Messages (SC28-7046)*.

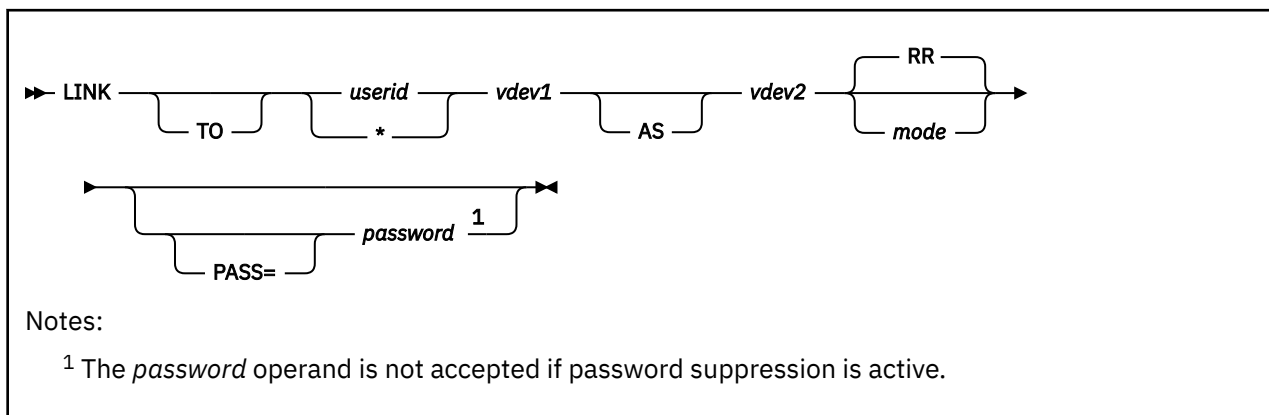
Messages:

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP013E Conflicting option - *option*
- HCP014E Conflicting operands - *operand1, operand2*
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP044E System {*name|volume*} does not exist
- HCP052E [XAUTOLOG failed for *userid*:] Error in CP directory
- HCP053E [XAUTOLOG failed for *userid*:] *userid/value* not in CP directory
- HCP059E {XAUTOLOG|AUTOLOG} failed for *userid* - {IPL missing|IPL failed|accounting errors|incorrect password}
- HCP170E System *name* exceeds storage size; {*nnnnnnnnK|nnnnM*} required
- HCP174E Paging I/O error; IPL failed
- HCP175E IPL is not allowed on a secondary CPU; IPL failed
- HCP177E Your PARM data exceeds the limit of nn characters. Correct your data string and reissue the IPL command.
- HCP232E IPL UNIT ERROR, {CSW=*csw*|IRB=*irb*}, SNS=*sense*
- HCP235E IPL with NSSDATA failed because the NSS {exceeds the end of the first extent of guest memory. | contains page range(s) of type SW, SC, or SN. | is defined with the VMGROUP attribute. | extends into unassigned increments.}
- HCP236E The NSSDATA operand is not allowed with the IPL of an NSS.
- HCP237E The NSSDATA operand must be specified after the DUMP keyword for the IPL of an FCP device.
- HCP260E IPL command processing cannot complete due to errors.
- HCP356E Access denied; User *userid* file *spoolid* not {changed|transferred|printed}



- HCP410E CP ENTERED; PAGING ERROR
- HCP447E System *sysname* cannot be IPLed due to an invalid processor configuration.
- HCP449E Your userid is not authorized to IPL system *name*.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP799E IPL failed due to insufficient or faulty storage
- HCP819I Traceid *traceid* disabled; group *group* enabled by *traceid*
- HCP838E The request process cannot complete due to a processor controller failure.
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*
- HCP1015E [XAUTOLOG failed for *userid*:] Insufficient storage is available to satisfy your request.
- HCP1122E *command* was not performed. Tape *vdev* has been given to user *userid*.
- HCP1352E System *name* cannot be processed.
- HCP1600E Specified cylinder number *cylinder* is not on the virtual DASD.
- HCP1601E The PARM option is not allowed for the system specified with the IPL command.
- HCP1612E Conflicting device numbers were specified on the IPL and SET LOADDEV or SET DUMPDEV commands.
- HCP1614E An IPL without the SECURE option cannot be performed because the system being dumped was IPLed with the SECURE option.
- HCP1614E An IPL without the SECURE option cannot be performed because the virtual machine has the SECUREIPLREQUIRED option set in its directory entry.
- HCP1615E IPL LOADDEV|DUMPDEV failed because not all necessary parameters have been set.
- HCP1616E IPL of a dump program failed because no dump save areas are available. Try again later.
- HCP1617E IPL by device number failed because secure IPL can be done only when using the LOADDEV or DUMPDEV form of the command.
- HCP2813E CP is running on a processor or in a mode that does not support IPL with DUMP to SCSI or ECKD devices.
- HCP2813E CP is running on a processor or in a mode that does not support IPL from SCSI or ECKD devices.
- HCP2815E The virtual storage size must be greater than *nnnnM* to perform a list-directed IPL.
- HCP2815E The virtual storage size of the CF Service Machine must be greater than *nnnnM* to load the message processor.
- HCP2815E The virtual storage size must be greater than *nnnnM* to perform a list-directed IPL with the DUMP option.
- HCP2816I Acquiring the machine loader from the processor controller.
- HCP2817I Load completed from the processor controller. Now starting the machine loader.
- HCP2818E The request to load the machine loader from the processor controller has been cancelled.
- HCP2824E No {LOADDEV|DUMPDEV} parameters are currently defined
- HCP6056I XAUTOLOG information for *userid*: The IPL command is verified by the IPL command processor.
- HCP6061I APPC wait canceled; virtual machine has been reset.
- HCP6086I Traceid *traceid1* is no longer superseded by VMGROUP trace *traceid2*.
- HCP6094I User *userid* will not be traced by trace *traceid*.
- HCP6097I User *userid* has restricted tracing to traces defined as BLOCK.
- HCP6525E Authorization request failed.

## LINK



## Authorization

Privilege Class: G

## Purpose

Use LINK to make a device associated with another virtual machine available to your virtual machine configuration.

## Operands

### *userid*

is the user ID in the system directory whose entry is to be searched for device *vdev1*. If you omit the TO operand, you cannot specify *userid* as T or TO.

\*

indicates that your devices are to be searched for device *vdev1*. When the LINK command is issued in an adjunct configuration, the asterisk refers to the directory entry for the issuing user, that is, the entry defining the user's principal configuration.

### *vdev1*

is the specified user's virtual device number, as defined in the user's entry in the system directory.

### *vdev2*

is the virtual device number that is to be assigned to the device for your virtual machine configuration.

### *mode*

is the link mode. Valid modes are:

#### Mode

##### Meaning

#### R

Read-only link. A read link is established, unless another user has an existing write or exclusive mode link to the disk.

#### RR

Read-only link. A read link is established, unless another user has an existing exclusive mode link. This is the default.

#### W

Write link. A write link is established, unless another user has an existing link (any mode) to the disk.

**WR**

Write link. A write link is established unless another user has an existing link (any mode) to the disk. If the write link is denied, a read link is established unless another user has an existing exclusive mode link for the disk.

**M**

Multiple-write link. A write link is established unless another user has an existing write, stable, or exclusive mode link to the disk.

**MR**

Multiple-write link. A write link is established unless another user has an existing write, stable, or exclusive mode link to the disk. In the case of a previous write or stable link, a read-only link is established. In the case of an exclusive mode link existing, the link is denied.

**MW**

Multiple-write link. A write link is established in all cases except when another user has an existing stable or exclusive mode link to the disk.

**Note:** A stable link means that the user holding the (SR, SW or SM) link can be assured that the disk remains stable, unchanged by others, until the link is released.

**SR**

Stable read-only link. A read link is established unless another user has an existing write or exclusive mode link to this disk. All requests for a write link to a disk with an existing SR mode link are denied.

**SW**

Stable write link. A write link is established unless another user has an existing link (any mode) to this disk. All requests for a write link to a disk with an existing SW mode link are denied.

**SM**

Stable multiple link. A write link is established unless another user holds either a write, a stable or an exclusive mode link to this disk. All requests for a write link to a disk with an existing SM mode link are denied.

**Note:** An EXCLUSIVE link (ER, EW), is a stable link with the added restriction that no one else has, or can get a link to, the specified minidisk until the link is released.

**ER**

Exclusive read-only link. A read link is established unless another user has an existing link to this disk. All requests for links to a disk with an existing exclusive mode link are denied.

**EW**

Exclusive write link. A write link is established unless another user has an existing link (any mode) to the disk. All requests for links to a disk with an existing exclusive mode link are denied.

If you do not specify a mode, the default is R if you are linking to another user's device. If you are linking to one of your own devices, the default is the user link mode specified in your entry in the system directory.

**Attention:**

- Permanent data loss can result if more than one user is writing to the same virtual disk.
- Unpredictable results can occur when one user has a read-only link to a disk that is being updated by another user who has a write link to the disk.

***password***

is a 1-character to 8-character string that matches the link mode password defined for device *vdev1* in the system directory entry for the user ID that you specify. If you do not specify a password but one is required, you are prompted for one.

To determine whether password suppression is active for this command, use the QUERY PASSWORD LINK command. If password suppression is active, you cannot specify a password on the command line. See usage note [“11” on page 580](#).

## Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.
2. When linking to another user's non-fullpack minidisk on a cached control unit, you are granted the cache access that is found in that user's directory.
3. When linking to another user's fullpack minidisk on a cached control unit, you get a subsystem control level of NOCTL. NOCTL means that no CCWs are accepted that can exercise control of subsystem resources or functions, regardless of whether they directly relate to the device. For example, subsystem storage may not be turned on or off, and caching may not be turned on or off, for the device.  
  
If you have a LINK statement for a fullpack minidisk in your directory, and DETACH the device after LOGON, you can get the device back with the same DASDOPT specified in the directory by LINK \* vdev1 AS vdev2. This is because this LINK is handled as a LINK to your own device.
4. CP verifies the LINK command operands you enter against the information contained in the other user's entry in the system directory. If the other user has temporarily redefined any of this information, you must still specify the information as it is indicated in the directory, or you receive an error message.
5. Do not confuse a link mode password with a user logon password. You specify a user logon password to gain access to the z/VM system. A link mode password controls whether you can link another user's DASD using the LINK command.
6. If the operand immediately following vdev2 is not a valid link mode, the default link mode is established, and the operand is interpreted as the link mode password. If you are using the default for the link mode, the link mode password you enter cannot be the same as any of the link modes (R, RR, W, WR, M, MR, MW, SR, SW, SM, ER, or EW).
7. If you are linking to one of your own disks, or if the link mode password is ALL (indicating that any user can use the disk), you do not have to enter the password. You can specify *userid* as an asterisk (\*) to indicate that the device is in your entry in the system directory; if you do specify a password, the system does not check it. If you have a LINK statement in your directory and an External Security Manager (ESM) is installed, the ESM may be called to see if you are in the access list for the minidisk.
8. If an ESM is installed on your system, you may not be authorized to enter the LINK command for all minidisks and all access modes. The ESM may downgrade certain requests for write access to read access. For additional information, contact your security administrator.
9. You cannot request virtual reserve/release processing using the LINK command. Use the MDISK directory statement to request virtual reserve/release processing. For more information about the [MDISK Statement](#), see *z/VM: CP Planning and Administration*.
10. The use of the stable and exclusive link modes (SR, SW, SM, ER, EW) is controlled by the LNKSTABL and LNKEXCLU options on the OPTION directory statement. For more information about the [OPTION Directory Statement](#), see *z/VM: CP Planning and Administration*.
11. If your installation is using the password suppression facility for this command (controlled by the PASSWORDS\_ON\_CMDs operand of FEATURES system configuration statement or by the SET PASSWORD command), you receive an error message if you enter the link mode password on the LINK command line. Wait for the ENTER PASSWORD prompt before you type in the password. The password is automatically masked, thereby improving system security.
12. In some responses and messages, *userid* refers to another virtual machine affecting the LINK command.
13. Messages are given to the user if:
  - a. A WRITE link exists to the requested disk or
  - b. A READ link exists to the disk and you are linking with mode WR, M, MR, or MW.

No messages are given if no other links exist, or a READ link exists, and you are linking with mode R or RR.

14. LINK response #2 is issued when CP does the link. If you issue the LINK command from CMS mode then no response is issued. If you issue LINK from CP READ mode, response #2 is issued. Also, if you issue #CP LINK, from either mode, response #2 is issued.
15. An informational response is not always generated after a read link is established.
16. LINK is used to obtain access directly to another user's minidisk, or to obtain access indirectly, through linking to another user's LINK or MDISK directory statement (LINKS to LINKS). Up to 50 of these *indirect* links (LINK indirections) are allowed. For example:

```

USER0
  MDISK 393 3390 1 10 USRVOL ALL ALL ALL
USER1
  LINK USER0 393 393 RR
USER2
  LINK USER1 393 393 RR
USER3
  LINK USER2 393 393 RR

```

Since USER1 has had to first access the 393 minidisk through a direct link to USER0, USER2's link to the 393 minidisk is considered an indirect link. USER3's link to the 393 minidisk through USER2 is thus also an indirect link. Up to 50 of these indirect links are allowed before the following message is issued:

**HCP109E**

*userid vdev* not linked; excessive LINK indirections

**Note:** This limit of 50 indirect links includes the original link. (In the example, USER1 to USER0.)

17. State change interrupts will not be reflected to guests who dynamically link to any fullpack minidisk. To receive state change interrupts for fullpack minidisks, a static link should be defined in the user directory followed by the DASDOPT statement specifying DEVCTL or SYSCTL level of control.
18. The link allowed by the LINK command to the *vdev1* device belonging to *userid* is summarized in the tables that follow.

Table 24. Linking to a Device in READ Mode

Link Mode You Entered	READ Password Specified in the System Directory?	Any Other Links to this Device?	Link Type You Will Establish
R	No	Not applicable	None
	Yes	No	READ
	Yes	Yes—READ, stable READ	READ
	Yes	Yes—WRITE, stable WRITE, exclusive (READ or WRITE)	None
SR	No	Not applicable	None
	Yes	No	Stable READ
	Yes	Yes—READ, stable READ	Stable READ
	Yes	Yes—WRITE, stable WRITE, exclusive (READ or WRITE)	None
ER	No	Not applicable	None
	Yes	No	Exclusive READ
	Yes	Yes—Any	None
RR	No	Not applicable	None
	Yes	No	READ
	Yes	Yes—READ, WRITE, stable (READ or WRITE)	READ
	Yes	Yes—Exclusive (READ or WRITE)	None

Table 25. Linking to a Device in WRITE Mode

Link Mode You Entered	WRITE Password Specified in the System Directory?	Any Other Links to this Device?	Link Type You Will Establish
W	No	Not applicable	None
	Yes	No	WRITE
	Yes	Yes—Any other link	None
EW	No	Not applicable	None
	Yes	No	Exclusive WRITE
	Yes	Yes—Any other link	None
SW	No	Not applicable	None
	Yes	No	Stable WRITE
	Yes	Yes—Any link	None
WR	No	Not applicable	None
	Yes	No	WRITE
	Yes	Yes—READ, WRITE, stable (READ, WRITE)	READ
	Yes	Yes—exclusive (READ, WRITE)	None

Table 26. Linking to a Device in Multiple WRITE Mode

Link Mode You Entered	MULT Password Specified in the System Directory?	Any Other Links to this Device?	Link Type You Will Establish
M	No	Not applicable	None
	Yes	No	WRITE
	Yes	Yes—READ	WRITE
	Yes	Yes—WRITE, stable (READ, WRITE), exclusive (READ, WRITE)	None
SM	No	Not applicable	None
	Yes	No	Stable WRITE
	Yes	Yes—READ	Stable WRITE
	Yes	Yes—WRITE, stable (READ, WRITE), exclusive (READ, WRITE)	None
MR	No	Not Applicable	None
	Yes	No	WRITE
	Yes	Yes—READ	WRITE
	Yes	Yes—WRITE, stable (READ, WRITE)	READ
	Yes	Yes—exclusive (READ, WRITE)	None

Table 26. Linking to a Device in Multiple WRITE Mode (continued)

Link Mode You Entered	MULT Password Specified in the System Directory?	Any Other Links to this Device?	Link Type You Will Establish
MW	No	Not Applicable	None
	Yes	No	WRITE
	Yes	Yes—READ, WRITE	WRITE
	Yes	Yes—stable (READ, WRITE), exclusive (READ, WRITE)	None

19. If the target minidisk is a virtual disk in storage that is defined in the specified user's directory entry but currently does not exist, the virtual disk in storage is created to satisfy your request, as long as it does not exceed the system limit on the storage available for virtual disks in storage. Note that a new virtual disk in storage must be formatted before it can be used.



**Attention:** Do not format an existing virtual disk in storage that you have linked to, or you will destroy all the data on it.

20. If you link a minidisk as virtual device number 192, the following special rules apply when you IPL CMS:
- If 192 is an unformatted virtual disk in storage, CMS formats it and accessed it as file mode D.
  - If 192 is a CP-formatted virtual disk in storage, CMS reformats it for CMS use and accesses it as file mode D.
  - If 192 is a CMS-formatted virtual disk in storage or permanent minidisk that is accessed as a file mode other than D, CMS reaccesses it as file mode D.
  - If 192 is an unformatted or CP-formatted permanent minidisk, CMS does not automatically format, reformat, access, or reaccess it.

When CMS accesses a 192 minidisk as file mode D, any minidisk or SFS directory already accessed as D is released.

21. If the directory definition of a permanent minidisk is changed while users are linked to it, existing links are unchanged, but any new links are made using the new definition of the minidisk. Unpredictable results may occur.
22. If the directory definition of a virtual disk in storage is changed while users are linked to it, existing links are unchanged, and any new links are made to the existing virtual disk in storage. The existing virtual disk in storage is used until the last user detaches it or logs off, at which time the virtual disk in storage is destroyed. After that, any new links use the new definition of the virtual disk in storage.
23. The CMS VMLINK command provides all the function of LINK and also allows you to specify a nickname for *userid vdev1*, finds a free virtual device number for *vdev2*, and also accesses the disk at a free mode letter. See *z/VM: CMS Commands and Utilities Reference*.
24. When linking to a minidisk that has at least one cylinder that is the target of a persistent FlashCopy relationship, warning message HCP116I is issued. The link is not prevented when a FlashCopy relationship exists.
25. Table 27 on page 584 shows how link conflicts are handled when virtual reserve/release is authorized and the minidisks have overlapping or matching extents. That is, the existing link and the requested link are to the same minidisk (one MDISK directory statement), or the links are to different minidisks (two MDISK directory statements) that partially or fully overlap.

Each link can be an explicit link obtained with a LINK directory statement or LINK command, or an instantiated link established for an MDISK directory statement when that user logs on.

Subsequent links will be analyzed individually for conflicts.

The following terms are used in this table:

## LINK

### Normal

means link conflict analysis is performed as documented in [Table 24 on page 581](#), [Table 25 on page 582](#), and [Table 26 on page 582](#).

### None

means link conflict analysis is not performed.

### V

means the minidisk is defined with access mode suffix V, which authorizes CP to use virtual reserve/release.

### 1-sys

means a non-SSI environment where both links are on the same system or an SSI cluster where both links are on the same member.

### 2-sys

means an SSI cluster where the existing link and the requested link are on different members.

### Convert to V

means the requested link is established with virtual reserve/release, although the minidisk is not defined with access mode suffix V.

Requested Link To	Existing Link To			
	Fullpack	Non-Fullpack	Fullpack V	Non-Fullpack V
Fullpack	Normal	None	<b>Pre-6.2:</b> Normal (V given to existing link but not to requested link) <b>1-sys and 2-sys:</b> Convert to V with message HCP6648I.	None
Non-Fullpack	None	Normal	<b>Pre-6.2:</b> None <b>1-sys and 2-sys:</b> Reject link.	<b>Pre-6.2:</b> Normal <b>1-sys:</b> Convert to V with message HCP6648I. <b>2-sys:</b> Reject link.
Fullpack V	<b>Pre-6.2:</b> Normal (V given to requested link but not to existing link.) <b>1-sys and 2-sys:</b> Reject link.	<b>Pre-6.2:</b> None <b>1-sys and 2-sys:</b> Reject link.	Normal	<b>Pre-6.2:</b> None <b>1-sys and 2-sys:</b> Reject link.
Non-Fullpack V	None	<b>Pre-6.2:</b> Normal (V given to requested link but not to existing link.) <b>1-sys and 2-sys:</b> Reject link.	<b>Pre-6.2:</b> None <b>1-sys and 2-sys:</b> Reject link.	<b>Pre-6.2:</b> Normal <b>1-sys:</b> Normal <b>2-sys:</b> Reject link.

26. Table 28 on page 585 shows how link conflicts are handled when simulated working allegiance is active and the minidisks have overlapping or matching extents. That is, the existing link and the requested link are to the same minidisk (one MDISK directory statement), or the links are to different minidisks (two MDISK directory statements) that partially or fully overlap.

Each link can be an explicit link obtained with a LINK directory statement or LINK command, or an instantiated link established for an MDISK directory statement when that user logs on.

Subsequent links will be analyzed individually for conflicts.



The following terms are used in this table:

**Normal**

means link conflict analysis is performed as documented in [Table 24 on page 581](#), [Table 25 on page 582](#), and [Table 26 on page 582](#).

**None**

means link conflict analysis is not performed.

**WRKALLEG**

means simulated working allegiance is active for the minidisk.

**Note:** If the links have already been established and you use the SET WRKALLEG command to turn on working allegiance, the new SET WRKALLEG becomes a requested link and is evaluated accordingly.

**1-sys**

means a non-SSI environment where both links are on the same system or an SSI cluster where both links are on the same member.

**2-sys**

means an SSI cluster where the existing link and the requested link are on different members.

Requested Link To	Existing Link To			
	Fullpack	Non-Fullpack	Fullpack WRKALLEG	Non-Fullpack WRKALLEG
Fullpack	Normal	None	<b>Pre-6.2:</b> Normal <b>1-sys:</b> Normal (HCP6648I issued to indicate WRKALLEG in effect.) <b>2-sys:</b> Reject link.	<b>Pre-6.2:</b> None <b>1-sys:</b> None <b>2-sys:</b> Reject link.

Table 28. Link Conflict Analysis When Simulated Working Allegiance is Active (continued)

Requested Link To	Existing Link To			
	Fullpack	Non-Fullpack	Fullpack WRKALLEG	Non-Fullpack WRKALLEG
Non-Fullpack	None	Normal	<b>Pre-6.2:</b> None <b>1-sys and 2-sys:</b> Reject link.	<b>Pre-6.2:</b> Normal (If extents match, both links have WRKALLEG in effect. If extents overlap but do not match, the existing link will have WRKALLEG in effect and the requested link will not.) <b>1-sys:</b> Normal (If extents match, both links have WRKALLEG in effect and message HCP6648I is issued for the link requested. If extents partially overlap, the existing link will have WRKALLEG in effect and the requested link will not.) <b>2-sys:</b> Reject link.
Fullpack WRKALLEG	<b>Pre-6.2:</b> Normal and neither link has WRKALLEG <b>1-sys:</b> Add WRKALLEG to existing link and issue unsolicited message HCP6648I to user with original link to indicate WRKALLEG has been added. <b>2-sys:</b> Reject link.	<b>Pre-6.2:</b> None <b>1-sys and 2-sys:</b> Reject link.	<b>Pre-6.2:</b> Normal <b>1-sys:</b> Normal <b>2-sys:</b> Reject link.	<b>Pre-6.2:</b> None <b>1-sys and 2-sys:</b> Reject link.

Requested Link To	Existing Link To			
	Fullpack	Non-Fullpack	Fullpack WRKALLEG	Non-Fullpack WRKALLEG
Non-Fullpack WRKALLEG	<b>Pre-6.2:</b> None <b>1-sys:</b> None <b>2-sys:</b> Reject link.	<b>Pre-6.2:</b> Normal. If extents match, the requested link does not get WRKALLEG. If extents partially overlap, the requested link gets WRKALLEG. <b>1-sys:</b> If extents match, add WRKALLEG to existing link and requested link and issue unsolicited message HCP6648I to user with original link to indicate WRKALLEG has been added. If extents overlap but do not match, the requested link will have WRKALLEG in effect and the existing link will not. <b>2-sys:</b> Reject link.	<b>Pre-6.2:</b> None <b>1-sys and 2-sys:</b> Reject link.	<b>Pre-6.2:</b> Normal <b>1-sys:</b> Normal <b>2-sys:</b> Reject link.

27. If the system has been IPLed in repair mode, a subconfiguration ID can be specified as the user ID on the LINK command.

## Responses

Response 1:

If your installation is using the password suppression facility, one of the following messages prompts you to enter the link mode password:

```
ENTER READ PASSWORD:
```

```
ENTER WRITE PASSWORD:
```

```
ENTER MULT PASSWORD:
```

Enter the appropriate password to obtain a link to the desired disk.

### Note:

1. There is a system threshold value that disables the LINK command when a user continues to enter trial passwords to obtain a link to a minidisk. The maximum is 10 incorrect passwords before the command is disallowed for the remainder of the logon session. The user is still able to link to the user's own minidisks or other's minidisks that have a password of ALL.
2. Responses 2 and 3 and redisplayed input lines are sent to the terminal only. They are not passed to the virtual machine when issued from the CMS environment or any environment which passes commands to CP by issuing a DIAGNOSE X'08'.

Response 2:

```
DASD vdev2 LINKED R/O
```

indicates that a read-only link has been established to the indicated disk with a mode of R, RR, SR or ER. To see all the links that exist for the minidisk, issue the QUERY LINKS command.

Response 3:

```
DASD vdev2 LINKED R/W
```

indicates that a read/write link has been established to the indicated disk with a mode of (W, WR, M, MR, MW, SW, SM or EW). To see all the links that exist for the minidisk, issue the QUERY LINKS command.

Response 4:

```
DASD vdev2 LINKED R/O; R/W BY {nnnnn USERS          }
                               {userid              }
                               {userid at sysname     }
                               {userid ADJUNCT        }
                               {userid ADJUNCT at sysname}

[;R/O BY {nnnnn USERS          }
         {userid              }
         {userid at sysname     }
         {userid ADJUNCT        }
         {userid ADJUNCT at sysname}]
```

indicates that a read-only link has been established to the indicated disk with a mode of RR, WR, or MR and warns that the disk is linked read/write by one or more users and possibly linked read-only by one or more users. If only one user has the indicated type of link, the number of links (*nnnnn* USERS) is replaced by that user's user ID or that user's user ID and the system name when the user ID is an instance of a multiconfiguration virtual machine on another system in the SSI cluster. To see all the links that exist for the minidisk, issue the QUERY LINKS command.

Response 5:

```
DASD vdev2 LINKED R/W; R/O BY {nnnnn USERS          }
                               {userid              }
                               {userid at sysname     }
                               {userid ADJUNCT        }
                               {userid ADJUNCT at sysname}
```

indicates that a read/write link has been established to the indicated disk with a mode of M, MR, MW, SW or SM and informs you that the disk is also linked read-only by one or more other users. No other read/write links to the disk exist. If only one user has a link, the number of links (*nnnnn* USERS) is replaced by that user's user ID or that user's user ID and the system name when the user ID is an instance of a multiconfiguration virtual machine on another system in the SSI cluster. To see all the links that exist for the minidisk, issue the QUERY LINKS command.

Response 6:

```
DASD vdev2 LINKED R/W; R/W BY {nnnnn USERS          }
                               {userid              }
                               {userid at sysname     }
                               {userid ADJUNCT        }
                               {userid ADJUNCT at sysname}

[;R/O BY {nnnnn USERS          }
         {userid              }
         {userid at sysname     }
         {userid ADJUNCT        }
         {userid ADJUNCT at sysname}]
```

indicates that a read/write link has been established to the given disk with a mode of MW, and warns you that the disk is linked read/write by one or more users and possibly linked read-only by some users. If only one user has a link, the number of links (*nnnnn* USERS) is replaced by that user's user ID. To see all the links that exist for the minidisk, issue the QUERY LINKS command.

Response 7:

When linking to minidisks that are in a FlashCopy relationship, you will see this response:

```
flashcopy establish source 200 0.10 target 300 50.10
Command complete: FLASHCOPY 0200 0 9 TO 0300 50 59
FLASHCOPY RELATIONSHIP 4AF7BDB3 ESTABLISHED
Ready; T=0.01/0.01 14:47:55

flashcopy establish source 300 0.10 target 200 50.10
Command complete: FLASHCOPY 0300 0 9 TO 0200 50 59
FLASHCOPY RELATIONSHIP 4AF7DB61 ESTABLISHED
Ready; T=0.01/0.01 14:48:02

detach 200 300
0200 0300 DETACHED
Ready; T=0.01/0.01 14:48:15

link * 200 200 w
HCPLNS116I WARNING: Cylinders 50 through 59 of minidisk 0200 are the
HCPLNS116I      target of FlashCopy relationship 4AF7DB61.
Ready; T=0.01/0.01 14:48:25

link * 300 300 w
HCPLNS116I WARNING: Cylinders 50 through 59 of minidisk 0300 are the
HCPLNS116I      target of FlashCopy relationship 4AF7BDB3.
Ready; T=0.01/0.01 14:48:29
```

## Messages

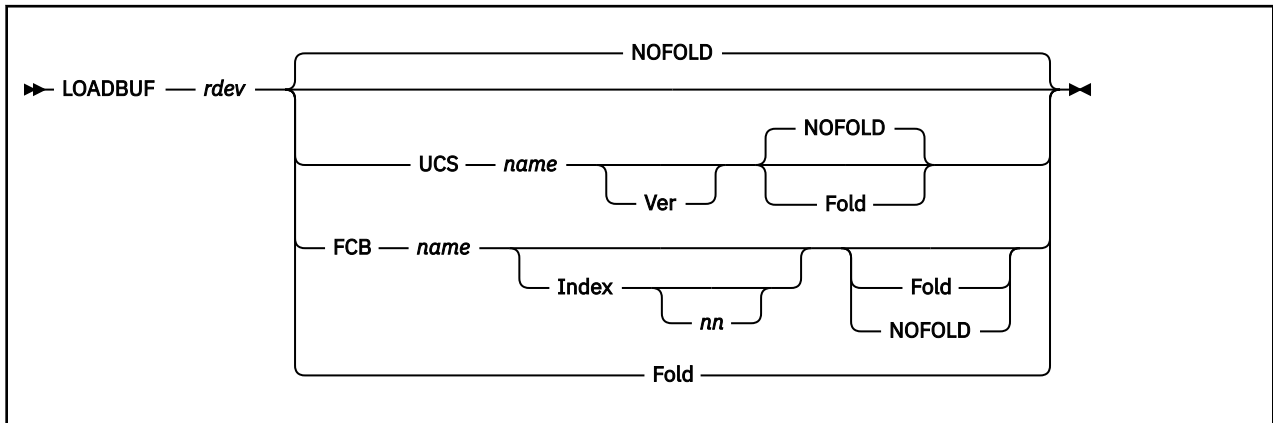
- HCP002E Invalid operand - *operand*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP024E Cylinder or block numbers missing or invalid
- HCP026E Operand missing or invalid
- HCP046E *type rdev* offline
- HCP052E Error in CP directory
- HCP053E *userid* not in CP directory
- HCP091E DASD *vdev* not defined; {temp|vdisk} space not available
- HCP092E Device *vdev* not defined; *device vdev* already defined
- HCP101E DASD *vdev* forced R/O; R/O by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}; stable by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}}
- HCP102E DASD *vdev* forced R/O; R/W by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}; stable by {userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}}
- HCP103E DASD *vdev* forced R/O; R/W by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}, R/O by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}; stable by {userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}}
- HCP104E *userid vdev* not linked; R/O by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}; stable by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}|exclusive by {userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}}
- HCP105E *userid vdev* not linked; R/W by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}; stable by {userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}|exclusive by {userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}}
- HCP106E *userid vdev* not linked; R/W by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}, R/O by {nnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname} [; stable by {userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname}}
- HCP107E *userid vdev* not linked; not in CP directory

## LINK

- HCP108E *userid vdev* not linked; valid *valid* not mounted
- HCP109E *userid vdev* not linked; invalid link device
- HCP110E *userid vdev* not linked; *type vdev* already defined
- HCP111E *userid vdev* not linked; no read password
- HCP112E *userid vdev* not linked; no write password
- HCP113E *userid vdev* not linked; no multiple password
- HCP114E *userid vdev* not linked; mode or password incorrect
- HCP115E {LINK|AUTOLOG|XAUTOLOG} invalid; excessive incorrect passwords
- HCP116I WARNING: Cylinders *n* through *n* of minidisk *vdev* are the target of FlashCopy relationship *hhhhhhh*.
- HCP117E Userid *userid* not linked; valid *valid* conflict
- HCP118E *userid vdev* not linked; command format not valid
- HCP153E Device {*vdev|range*} exceeds maximum number of virtual devices
- HCP298E *userid vdev* not linked; request denied
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP844E The device is not defined due to an invalid control unit type.
- HCP1150E DASD {*rdev|vdev*} is not a valid base exposure.
- HCP1151E *userid vdev* has not been linked because it is not within CP volume cylinder extents.
- HCP1152E *userid vdev* has not been linked because it would overlap system {*spool|paging|temporary disk*} space
- HCP1153E *userid vdev* cannot be re-linked {R/O|R/W} because it is part of a minidisk-pool and was previously linked {R/W|R/O}.
- HCP1155E You do not have the proper directory authorization to LINK a minidisk with the link mode of *mode*.
- HCP1156I DASD *device number* forced R/O; unauthorized for R/W
- HCP1157I DASD *vdev* was not LINKed because the MDISK DEVNO parameter requires the device to be a free volume.
- HCP1158E Stable and exclusive links are not supported. A system that shares this volume is not at VM/ESA 1.1 or later.
- HCP1159E DASD *address* is not linked; RDEV *rdev* is already attached.
- HCP2010E DASD *dasd* is a full-pack mini on a shared real volume, WRKALLEG is rejected.
- HCP2880E *userid vaddr* is not linked, CSE LINK I/O error, code *cc*.
- HCP6525E Authorization request failed.
- HCP6640E *userid vdev* not linked. DASD is a VDISK and is already linked on system *sysname*.
- HCP6640E *userid vdev* not linked. DASD is a VDISK and is defined under a SUBCONFIG for another SSI member.
- HCP6640E *userid vdev* not linked. MDISK DEVNO parameter conflicts with minidisk linked by {*userid|userid* at *sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk has been defined with the V mode suffix and is already linked by {*userid|userid* at *sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk has been defined with WRKALLEG and is already linked by {*userid|userid* at *sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk is already linked with the V mode suffix by {*userid|userid* at *sysname*}.
- HCP6640E *userid vdev* not linked. Minidisk is already linked with WRKALLEG in effect by {*userid|userid* at *sysname*}.

- HCP6640E *userid vdev* not linked. Minidisk is linked with the MDISK DEVNO parameter by {*userid|userid* at *sysname*}.
- HCP6640E *userid vdev* not linked. No EQID exists for RDEV *rdev*.
- HCP6640E *userid vdev* not linked. SSI cluster error, code *code*.
- HCP6645E *userid vdev* not linked; SSI cluster is in safe mode.
- HCP6648I MDISK *vdev* has virtual reserve/release support in effect.
- HCP6648I MDISK *vdev* has working allegiance simulation in effect.
- HCP6761E *userid vdev* not linked; volume *valid* marked draining by Operations personnel.

## LOADBUF



### Authorization

Privilege Class: D

### Purpose

Use LOADBUF to load the following:

- The universal character set (UCS) buffer with a specified print chain/train image for the IBM 3203 model 5 printer, 3262 printer, or a device that emulates a 3211 printer.
- The forms control buffer (FCB) with a specified image for the IBM 3203 model 5, 3262, 4245, 4248 printer, or a device that emulates a 3211 printer.

### Note:

1. You cannot load a UCS for a 4245 or 4248 printer.
2. The INDEX option is valid only for devices that emulate the 3211 printer.
3. To load the forms control buffer (FCB) with a specified image for a **virtual printer**, use the LOADVFCB command.
4. The number of lines to be printed on the separator page is determined by the FCB loaded on the printer. The default separator page contains 58 lines of data. If the page length defined by the FCB is less than the default separator page length, the separator page data must be customized in order to fit on a single page. This may be done using CP Exit points that are provided in separator page processing. See [z/VM: CP Exit Customization](#) for details.

### Operands

#### rdev

indicates the real device number of the printer whose buffer is to be loaded.

#### NOFOLD

#### Fold

If FOLD is specified, all characters will be printed in uppercase.

If NOFOLD is specified, the characters will not be not folded into uppercase.

If neither FOLD nor NOFOLD is specified with the UCS option, then NOFOLD is the default. FOLD is used when a file specifies lowercase, but no lowercase UCS is available.

If neither FOLD nor NOFOLD is specified with the FCB option, then the FOLD/NOFOLD selection on the printer will not be changed.



**Note:** If you specify the FOLD option, the control unit ignores bit positions 0 and 1 of the EBCDIC code when comparing the character codes from the UCS buffer and the print line buffer. Only uppercase characters are printed from either uppercase or lowercase data codes.



**Attention:** Misuse of the FOLD option can cause multiple characters to be printed for the same EBCDIC bit configuration.

**UCS name**

**FCB name**

specifies that you are loading a print chain image (UCS) or a forms control buffer (FCB). If you are loading both a print chain image and a forms control buffer, you must enter two LOADBUF commands for the specified printer.

The operand *name* is the 1- to 4-character name of the UCS or FCB image that you are loading. The IBM-supplied names are given in the tables that follow. IBM supplies the following UCS buffer images for the 3203 model 5 printer:

**Name**

**Meaning**

**AN**

Normal alphanumeric notation character set arrangement

**HN**

Normal hexadecimal notation character set arrangement

**PCAN**

Preferred alphanumeric notation character set arrangement

**PCHN**

Preferred hexadecimal notation character set arrangement

**QN**

PL/I—60 graphics

**QNC**

PL/I—60 graphics

**RN**

FORTRAN, COBOL commercial

**YN**

High-speed alphanumeric

**TN**

Text printing 120 graphics

**PN**

PL/I—60 graphics

**SN**

Text printing 84 graphics

IBM supplies the following UCS buffer images for a device that emulates the 3211 printer:

**Name**

**Meaning**

**A11**

Standard commercial

**H11**

Standard scientific

**G11**

ASCII

**P11**

PL/I

## LOADBUF

### T11

Text printing

IBM supplies the following UCS buffer images for the 3262 printer:

Name	Meaning
------	---------

<b>P48</b>	48-character belt
------------	-------------------

<b>P52</b>	52-character belt (Austria/Germany)
------------	-------------------------------------

<b>P63</b>	63-character belt, optimized
------------	------------------------------

<b>P64</b>	64-character belt
------------	-------------------

<b>P96</b>	96-character belt
------------	-------------------

<b>P116</b>	116-character belt (French-Canadian)
-------------	--------------------------------------

<b>P128</b>	128-character belt (Katakana)
-------------	-------------------------------

IBM supplies the following FCB images for the 3203 Model 5, 3262, 4245, 4248 printers, and devices that emulate a 3211 printer:

Name	Meaning
------	---------

<b>FCB1</b>	Space 6 lines/inch, Length 66 lines per page
-------------	--

Line	Channel Skip Specification
------	----------------------------

<b>1</b>	1
----------	---

<b>3</b>	2
----------	---

<b>5</b>	3
----------	---

<b>7</b>	4
----------	---

<b>9</b>	5
----------	---

<b>11</b>	6
-----------	---

<b>13</b>	7
-----------	---

<b>15</b>	8
-----------	---

<b>19</b>	10
-----------	----

<b>21</b>	11
-----------	----

<b>23</b>	12
-----------	----

**64**

9

**Name****Meaning****FCB8**

Space 8 lines/inch, Length 68 lines per page

**Line****Channel Skip Specification****1**

1

**4**

2

**8**

3

**12**

4

**16**

5

**20**

6

**24**

7

**28**

8

**32**

10

**36**

11

**63**

12

**66**

9

**Name****Meaning****FCBS**

Space 8 lines/inch, Length 68 lines per page

**Line****Channel Skip Specification****1**

1

**54**

2

**55**

3

**56**

4

**57**

5

## LOADBUF

<b>58</b>	6
<b>59</b>	7
<b>60</b>	8
<b>61</b>	10
<b>62</b>	11
<b>63</b>	12
<b>64</b>	9

For the exact contents of these buffer images, see *IBM 3211 Printer, 3216 Interchangeable Train Cartridge*, and *3811 Printer Control Unit Component Description and Operator's Guide*.

For information on how to add more FCB and UCS images, see [Creating and Modifying Image Libraries for Printers in z/VM: CP Planning and Administration](#).

### Ver

prints the contents of the buffer on the specified printer. This lets you verify visually that the correct buffer was loaded.

### Index *nn*

starts printing the output in the designated (*nn*) print position. The indexing value *nn* must be a number from 0 through 31; a leading zero need not be specified. If you do not specify INDEX or if you specify an INDEX of 0 then indexing is turned off and no indexing is performed.

If you specify the INDEX keyword without an indexing value then the value coded in the FCB becomes the indexing value. If there is no value coded in the FCB then message 036E is issued. If you specify an indexing value from 1 to 31 then that value overrides the value in the FCB and is used as the indexing value.

Note that the INDEX option is valid only for a device emulating a 3211 printer.

## Usage Notes

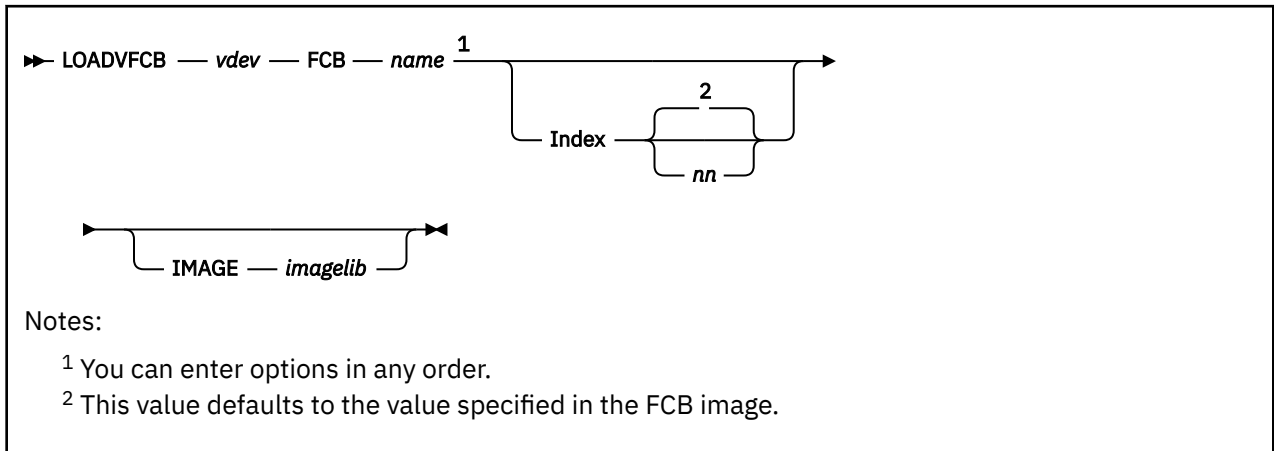
1. You must drain the printer before you enter the LOADBUF command.
2. The operator who is authorized to load the UCS and FCB buffers should use this command whenever:
  - The printer or control unit is powered on
  - The print chain or train is to be changed
  - The buffer has been causing an excessive amount of parity errors
  - The printer was previously attached to a user's virtual machine.
3. Load the FCB buffer whenever:
  - The FCB buffer is to be changed
  - The printer was previously attached to a user's virtual machine.
4. The LOADBUF command automatically sets the Block Data Check latch for a UCS buffer load. The Block Data Check latch is also set when the FOLD/NOFOLD option is selected for the LOADBUF command.
5. If you specify VER, the image of the UCS load will be printed on the printer you indicated. The printed output must correspond to the description of the buffer load in the publications:
  - *IBM 2821 Control Unit*

- *IBM 3203 Printer model 5 Component Description and Operator's Guide*
- *IBM 3211 Printer, 3216 Interchangeable Train Cartridge*
- *IBM 3262 Printer model 1 and 11 Component Description*
- *IBM 3811 Printer Control Unit Component Description and Operator's Guide.*

## Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option* }
- HCP021E A real device number was not supplied or it is invalid.
- HCP031E Buffer missing or invalid; [FCB too long|extended FCB not supported]
- HCP036E Index missing or invalid
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP046E *type rdev* offline
- HCP140E *type {rdev/ldev}* attached to *userid*
- HCP142E *type rdev* not drained
- HCP148E Printer *rdev* buffer load *name* failed
- HCP241E An undefined load image *imagename* was specified for printer *rdev*.
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP418E *{rdev/vdev}* is not a spooled device.
- HCP419E *{rdev/vdev}* is not a spooled printer.
- HCP421E Printer *prt-type* does not support the *{feature feature|command command}*. [*UCS\_name* is the name of the UCS associated with the currently installed print band|The UCS could not be loaded or read]
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*
- HCP1562E The length of buffer *bufname* is invalid.
- HCP2500E Image library *libname* could not be found [for printer *vdev*].
- HCP2501E A spool paging I/O error occurred while reading image [library] *imagename* on printer *rdev*.

## LOADVFCB



### Authorization

Privilege Class: G

### Purpose

Use LOADVFCB to specify the forms control buffer image for a virtual impact printer.

**Note:** To load the forms control buffer (FCB) with a specified image for a *real* printer, use the LOADBUF command.

### Operands

#### *vdev*

identifies the virtual device number of your virtual spooled impact printer.

#### **FCB *name***

specifies the forms control buffer image for your virtual printer. The operand *name* is the 1- to 4-character name of the FCB image. You may specify either the name of an IBM-supplied FCB image or the name of another FCB image supplied by your system programmer. The three IBM-supplied FCB images are FCB1, FCB8, and FCBS. The default FCB image is FCB1. The formats of the IBM-supplied FCB images are as follows:

#### **Name**

##### **Meaning**

#### **FCB1**

Space 6 lines/inch, Length 66 lines per page

#### **Line**

##### **Channel Skip Specification**

**1**

1

**3**

2

**5**

3

**7**

4

**9**  
 5  
**11**  
 6  
**13**  
 7  
**15**  
 8  
**19**  
 10  
**21**  
 11  
**23**  
 12  
**64**  
 9

**Name**  
**Meaning**

**FCB8**  
 Space 8 lines/inch, Length 68 lines per page

**Line**  
**Channel Skip Specification**

**1**  
 1  
**4**  
 2  
**8**  
 3  
**12**  
 4  
**16**  
 5  
**20**  
 6  
**24**  
 7  
**28**  
 8  
**32**  
 10  
**36**  
 11  
**63**  
 12  
**66**  
 9

**Name**  
**Meaning**

**FCBS**

Space 8 lines/inch, Length 68 lines per page

**Line****Channel Skip Specification**

<b>1</b>	1
<b>54</b>	2
<b>55</b>	3
<b>56</b>	4
<b>57</b>	5
<b>58</b>	6
<b>59</b>	7
<b>60</b>	8
<b>61</b>	10
<b>62</b>	11
<b>63</b>	12
<b>64</b>	9

For the exact contents of these FCB images, see *IBM 3211 Printer, 3216 Interchangeable Train Cartridge*, and *3811 Printer Control Unit Component Description and Operator's Guide*.

**Index *nn***

identifies the first print position. The operand *nn* must be a number from 1 through 31; a leading zero need not be specified. If you enter the keyword INDEX without a value, the value defaults to the value specified in the FCB image.

Note that the INDEX option is valid only for a virtual spooled 3211 printer; it is not valid for other impact printers.

**IMAGE *imagelib***

specifies the image library that contains the specified FCB. If you do not specify the IMAGE option, you get the default image library for the device type. The default image library is IMAGxxxx, where xxxx is the device type; for example, the default image library is IMAG1403 for a 1403 type printer.

**Usage Notes**

1. The virtual FCB image should match the FCB image on the real printer. If it does not, results are unpredictable.  
If the spool file is printed on a real 1403 printer, the virtual FCB image is ignored, and channel skipping is controlled by the carriage tape.
2. If you are running a program in your virtual machine that requires printer overflow sensing information and you are printing on a real 1403, you need to take special steps to retrieve the sensing information. First, ensure that the CP directory entry for your virtual machine indicates that the printer is a 3203, 3211, 3262, 4245, or 4248, even though the program and the guest operating system in which the



program runs define the printer as a 1403. Then use the LOADVFCB command to specify a virtual FCB image that matches the 1403. CP uses this virtual FCB image as a basis for retrieving sensing information.

3. If the virtual machine operating system issues a LOAD FCB channel command word to the virtual spooled printer, the LOAD FCB CCW image replaces the virtual FCB image provided on the LOADVFCB command line. Note that the CMS PRINT command issues a LOAD FCB CCW and thereby replaces the virtual FCB image.
4. The LOADVFCB command allows specification of the extended FCB image for a virtual 4248 printer. If an attempt is made to send the extended FCB image to a printer that does not support the extended FCB format, an error message results.
5. If the default FCB image has been modified, the virtual FCB image from the spooled printer is stored in any file created on that printer. When the file is printed on a real printer, the FCB is loaded into the real printer according to the FILEFCB or DEFFCB option specified on the START command. The file then prints properly on the real printer.

## Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP031E Buffer missing or invalid; [FCB too long|extended FCB not supported]
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP241E An undefined load image *imagename* was specified for printer *vdev*.
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP418E *{rdev/vdev}* is not a spooled device.
- HCP419E *{rdev/vdev}* is not a spooled printer.
- HCP421E Printer *pvt-type* does not support the *{feature feature|command command}*.
- HCP422E The same option was specified twice.
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*
- HCP1562E The length of buffer *bufname* is invalid.
- HCP2500E Image library *libname* could not be found [for printer *vdev*].
- HCP2501E A spool paging I/O error occurred while reading image [library] *imagename* on printer *vdev*.

## LOCATE

---

### Authorization

Privilege Class: C, E

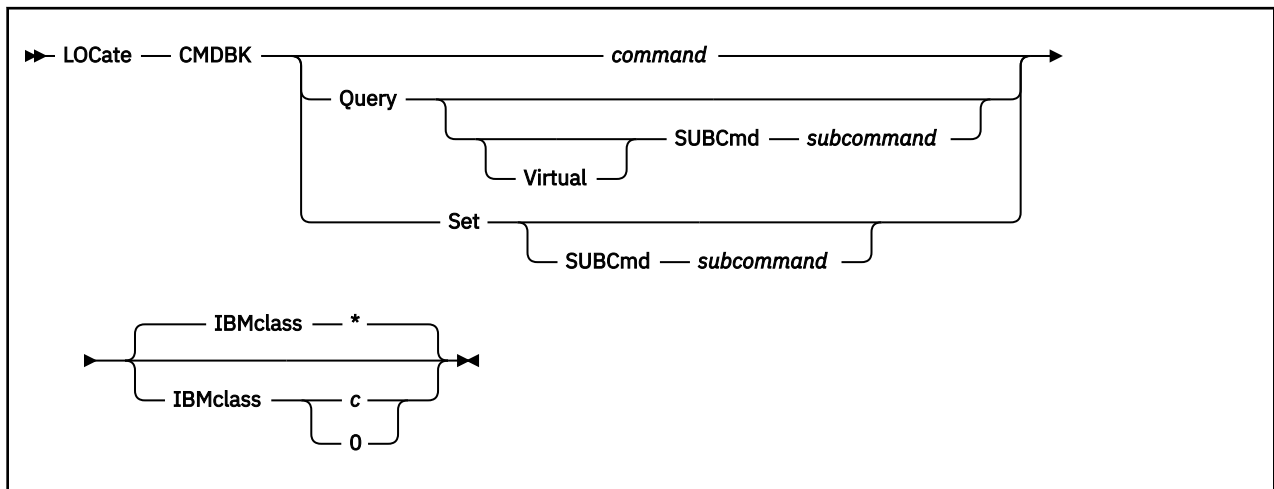
### Purpose

The LOCATE command is provided to enable you to perform CP problem diagnosis.

Use LOCATE to display the addresses of control information within CP. You can use the LOCATE command to find the addresses of:

- A command's CP command table entry block (CMDBK)
- A DIAGNOSE code's diagnose control block (DGNBK)
- A user's spool file identification entry (FILID)
- A system frame table entry (FRAMETBL)
- An exit point's indirect call locator block (ICLBK)
- A system logical device block (LDEV)
- A system real device block (RDEV)
- A real PCI function block (RPCI)
- A system shared system name block (SHPBK)
- A system SNA control block (SNABK)
- A user's spool file block (SPFBK)
- Data within host storage (Storage)
- A system execution space page management table entry (SXSTE)
- A system external symbol address (SYMBOL)
- A user's virtual device block (VDEV)
- A user's virtual machine definition block (VMDBK)
- A user's virtual PCI function block (VPCI)
- A system VTAM service machine block (VSMBK)
- An exit point's exit control block (XITBK).

## LOCATE CMDBK



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE CMDBK to display the address of the CP command table entry block (CMDBK) for a specified CP command.

### Operands

#### *command*

tells CP to locate the command table entry blocks for the specified CP command. The variable *command* is any valid CP command, any valid abbreviation of that command, or any character string (including the pattern matching characters, \* and %) representing that command. For more information about minimum abbreviations, see Usage Note “2” on page 604.

#### Query

tells CP to locate the command table entry blocks for the root CP QUERY command. To locate the command table entry blocks for a specific QUERY subcommand, specify the SUBCMD operand.

#### Virtual

tells CP that you are locating a QUERY VIRTUAL subcommand.

#### SUBCmd *subcommand*

specifies the name of the QUERY or SET subcommand whose command table entry blocks you want to locate. The variable *subcommand* is any valid QUERY or SET subcommand, any valid abbreviation of that subcommand, or any character string (including the pattern matching characters, \* and %) representing that subcommand. For more information about minimum abbreviations, see Usage Note “2” on page 604.

#### Set

tells CP to locate the command table entry blocks for the root CP SET command. To locate the command table entry blocks for a specific SET subcommand, specify the SUBCMD operand.

#### IBMclass \*

specifies that you want to find the command table entry blocks for all the IBM classes associated with the specified CP command. If omitted, IBMCLASS \* is the default.

**IBMclass c**

specifies that you only want to find the command table entry blocks for the specific IBM class associated with the specified CP command. The variable *c* is the IBM-defined privilege class A, B, C, D, E, F, or G.

**IBMclass 0**

specifies that you only want to find the command table entry blocks for CP commands with no assigned IBM class. Only commands with privilege class "ANY" have no assigned IBM class.

**Usage Notes**

1. For each command, CP has at least one command table entry block. If the command has more than one IBM class, CP has one command table entry block for each version of the command. For example, the DEFINE command has three IBM classes: A, E, and G. To locate the command table entry block for the DEFINE GRAF command, you want the command table entry block for the class G DEFINE command:

```
locate cmdbk define ibmclass g
```

The only exceptions to this are the QUERY and SET commands. CP has at least one command table entry block for each QUERY and SET subcommand. To locate the command table entry block for the QUERY TIMEZONES subcommand, you would issue one of the following commands:

```
locate cmdbk query subcmd timezones
locate cmdbk query subcmd timezones ibmclass g
```

2. You can use the minimum abbreviation of the command or subcommand name. For example, issuing either of the following commands produces the same results:

```
locate cmdbk query subcmd privclass
locate cmdbk q subcmd priv
```

3. To define a new CP command or a new version (by IBM class) of an existing CP command, use the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD”](#) on page 143).
4. To define a new alias for an existing CP command, use the DEFINE ALIAS command (see [“DEFINE ALIAS”](#) on page 127).
5. To load the command processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD”](#) on page 111).
6. To activate a CP command:
  - While defining it, use the ENABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD”](#) on page 143).
  - After defining it, use the ENABLE COMMAND or CMD command (see [“ENABLE COMMAND / CMD”](#) on page 460).
7. To change the definition of an existing CP command, use the MODIFY COMMAND command (see [“MODIFY COMMAND / CMD”](#) on page 683).
8. To deactivate a CP command:
  - While defining it, use the DISABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD”](#) on page 143).
  - After defining it, use the DISABLE COMMAND or CMD command (see [“DISABLE COMMAND / CMD”](#) on page 313).
9. To remove the command processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117).
10. For more information about user-defined commands, see [z/VM: CP Exit Customization](#).

## Responses

Response 1:

Command	CMDBK	Alias	IBMCLASS	Privilege
<i>command</i>	<i>nnnnnnnn</i>	<i>alias</i>	<i>c</i>	<i>p</i>
:				

### Command

shows the command name you specified, or a command name that matches the pattern you specified. When locating a QUERY or SET subcommand, the column header changes to **Subcommand** and the rows of data contain the name of the subcommand. See Example 5.

### CMDBK

shows the host logical storage address of the command table entry block for that command.

### Alias

tells you whether the command is an alias for another command. The possible values are:

**\***

this command is not an alias for any other command.

**hhhhhhhh**

this command is an alias for the command whose CMDBK is located at host logical storage address *hhhhhhhh*.

### IBMCLASS

shows the IBM class of the command. The possible values are:

**A**

this is a system-control command to be used by the primary system operator.

**B**

this is a command for operational control of real devices.

**C**

this is a command to alter system storage.

**D**

this is a command for system-wide control of spool files.

**E**

this is a command to examine system storage.

**F**

this is a command for service control of real devices.

**G**

this is a general-use command used to control the functions of a virtual machine.

**0**

(zero) this command has no IBM class.

**-**

this is an alias to the command located at the specified command table entry block address. As an alias, it has the same IBM class as the existing command. Thus, the information is not duplicated here.

### Privilege

shows the privilege class (or classes) of the command. This column usually contains one or more privilege classes in the range A to Z and 1 to 6. This column may also contain:

**<ANY>**

this command can be issued by users with any privilege class.

**-**

this is an alias to the command located at the specified command table entry block address. As an alias, it has the same privilege classes as the existing command. Thus, the information is not duplicated here.

**0**

this command has no privilege classes.

**Examples****Example 1:**

To find the command table entry block addresses for the CP MESSAGE command, enter the following:

```
locate cmdbk message
Command  CMDBK   Alias   IBMCLASS Privilege
MESSAGE  00344858 *       A       A
MESSAGE  003448A0 *       B       B
MESSAGE  003448E8 *       0       <ANY>
Ready;
```

**Example 2:**

The CP MSG command is actually an alias for the CP MESSAGE command. CP displays different information for an alias than it does for a command. For example, to find the command table entry block addresses for the CP MSG command alias, enter the following:

```
locate cmdbk msg
Command  CMDBK   Alias   IBMCLASS Privilege
MSG      00344A50 00344858 -       -
MSG      00344A98 003448A0 -       -
MSG      00344AE0 003448E8 -       -
Ready;
```

**Example 3:**

The CP MESSAGE and MSG commands also share the same minimum abbreviation, M. This means you can display information about both commands by specifying their common minimum abbreviation:

```
locate cmdbk m
Command  CMDBK   Alias   IBMCLASS Privilege
MESSAGE  00344858 *       A       A
MESSAGE  003448A0 *       B       B
MESSAGE  003448E8 *       0       <ANY>
MSG      00344A50 00344858 -       -
MSG      00344A98 003448A0 -       -
MSG      00344AE0 003448E8 -       -
Ready;
```

**Example 4:**

To find the command table entry block addresses for all CP commands whose name begins with the letter M, enter the following:

```
locate cmdbk m*
Command  CMDBK   Alias   IBMCLASS Privilege
MESSAGE  00344858 *       A       A
MESSAGE  003448A0 *       B       B
MESSAGE  003448E8 *       0       <ANY>
MIGRATE  00344930 *       A       A
MONITOR  00344978 *       A       M
MONITOR  003449C0 *       E       S
MSGNOH   00344A08 *       B       KRZ
MSG      00344A50 00344858 -       -
MSG      00344A98 003448A0 -       -
MSG      00344AE0 003448E8 -       -
Ready;
```

In this example, note that the classes for the MONITOR and MSGNOH commands were redefined by this installation through user MODIFY commands.

**Example 5:**

To find the command table entry block addresses for the SET PRIVCLASS subcommand, enter the following:

```
locate cmdbk set subcmd privclass
Subcommand  CMDBK      Alias      IBMCLASS  Privilege
PRIVCLASS   003B00C0 *          C          C
PRIVCLASS   003B0108 *          0         <ANY>
Ready;
```

### Example 6:

If someone had used a MODIFY COMMAND command or configuration file statement to override an existing CP DISABLE command, you would see something like this:

```
locate cmdbk disable
Command     CMDBK      Alias      IBMCLASS  Privilege
DISABLE     1251EF80 *          A          A
DISABLE     1131EE80 *          B          Z6
-----
DISABLE     12535560 *          B          B
DISABLE     11140780 *          F          F
Ready;
```

This response shows you there are command table entry blocks for four DISABLE commands: one for IBM class A, two for IBM class B, and one for IBM class F. The first IBM class B DISABLE command is the one that is active. The second IBM class B DISABLE command was the original IBM class B command.

To return to the original IBM class B DISABLE command, enter the following:

```
modify command disable ibmclass b reset
```

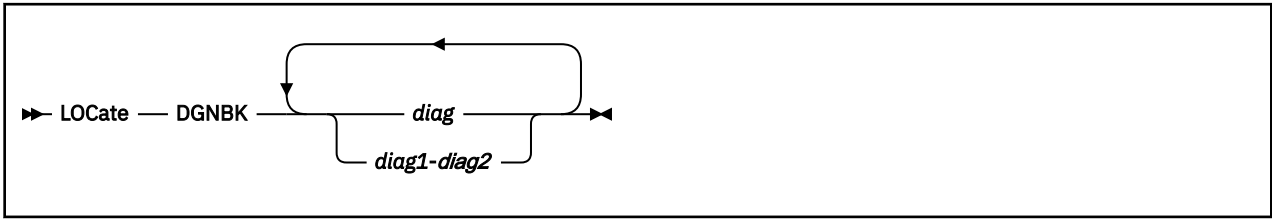
This causes the first IBM class B DISABLE command to disappear from the display and changes the ----- to DISABLE. Thus, the response would look like this:

```
locate cmdbk disable
Command     CMDBK      Alias      IBMCLASS  Privilege
DISABLE     1251EF80 *          A          A
DISABLE     12535560 *          B          B
DISABLE     11140780 *          F          F
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP6704E Missing token at end of line
- HCP6706E Invalid command name - *command*
- HCP6706E Invalid IBM class - *c*
- HCP6706E Invalid subcommand name - *subcommand*
- HCP6720I No items found to satisfy selection criteria.
- HCP6840E You specified QUERY VIRTUAL without SUBCMD

## LOCATE DGNBK



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE DGNBK to display the address of the CP DIAGNOSE code control block (DGNBK) for a specific DIAGNOSE code.

### Operands

*diag*

*diag1-diag2*

is the number of the DIAGNOSE code whose control block address you want to display. Each *diag* must be an existing DIAGNOSE code between X'0000' and X'03FC' and must be a multiple of 4. You can specify a single DIAGNOSE code, a range of DIAGNOSE codes, or any combination thereof.

### Usage Notes

1. To define a new DIAGNOSE code, use the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE” on page 172](#)).  
If you do not specify the ENABLE operand, a new DIAGNOSE code is initially in a disabled state after being defined. CP treats disabled DIAGNOSE codes as if they were never defined. If you try to use a disabled DIAGNOSE code in a program, CP will give you a program check specification exception.
2. To load the DIAGNOSE processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD” on page 111](#)).
3. To activate a new DIAGNOSE code:
  - While defining it, use the ENABLE operand of the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE” on page 172](#)).
  - After defining it, use the ENABLE DIAGNOSE command (see [“ENABLE DIAGNOSE” on page 463](#)).
4. To change the definition of an existing DIAGNOSE code after initialization, use the MODIFY DIAGNOSE command (see [“MODIFY DIAGNOSE” on page 697](#)).
5. To display information about a DIAGNOSE code (status, entry point name, and privilege class), use the QUERY DIAGNOSE command (see [“QUERY DIAGNOSE” on page 885](#)).
6. To deactivate a DIAGNOSE code:
  - While defining it, use the DISABLE operand of the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE” on page 172](#)).
  - After defining it, use the DISABLE DIAGNOSE command (see [“DISABLE DIAGNOSE” on page 316](#)).
7. To remove the DIAGNOSE processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD” on page 117](#)).



8. Many external security managers (ESMs) do not support DIAGNOSE codes above X'03FC'. For this reason, CP does not support DIAGNOSE codes above X'03FC'. The DIAGNOSE codes between X'0000' and X'03FC' are divided as follows:

**X'0000' to X'00FC'**

Reserved for IBM use

**X'0100' to X'01FC'**

Reserved for customer use

**X'0200' to X'03FC'**

Reserved for IBM use.

9. For more information about user-defined DIAGNOSE codes, see the [Introduction](#) in *z/VM: CP Exit Customization*.

## Responses

Response 1:

```
Code DGNBK
code nnnnnnnn
:
```

### Code

shows the DIAGNOSE code you specified, or a DIAGNOSE code within the range you specified.

### DGNBK

shows the host logical storage address of the DIAGNOSE code control block for that DIAGNOSE code.

## Examples

Example 1:

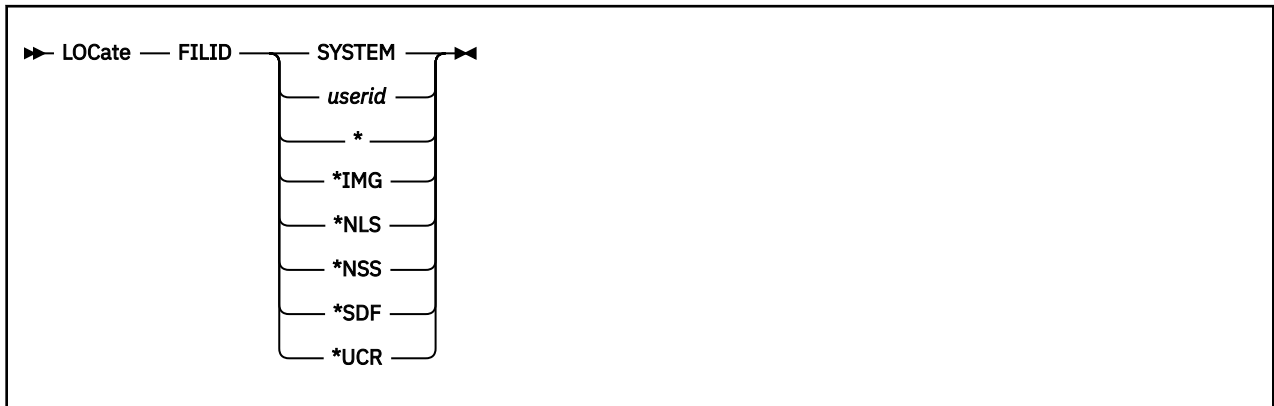
To display the address of the DIAGNOSE code table entry for DIAGNOSE code X'0C', enter the following:

```
locate dgnbk c
Code DGNBK
000C 005F09C8
Ready;
```

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP009E Invalid range - *range*
- HCP026E Operand missing or invalid
- HCP2751E Diagnose code *diag* does not exist
- HCP2751E Diagnose codes *diag1-dia2* do not exist
- HCP6706E Invalid Diagnose code - *diag*

## LOCATE FILID



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE FILID to display the address of a user's spool file identification (FILID) entry.

### Operands

#### SYSTEM

tells CP to locate the FILID entries for all user IDs on your system.

#### *userid*

tells CP to locate the FILID entries for the user ID, user ID pattern, or pseudo-user ID (\*IMG, \*NLS, \*NSS, \*UCR, or \*SDF) specified. For more information on user ID patterns, see Usage Note [“1”](#) on [page 610](#).

**\***

tells CP to locate the FILID entry for your user ID.

#### **\*IMG**

tells CP to locate the FILID entry for all printer image library files owned by the system.

#### **\*NLS**

tells CP to locate the FILID entry for all National Language Support files owned by the system.

#### **\*NSS**

tells CP to locate the FILID entry for all named saved system files owned by the system.

#### **\*SDF**

tells CP to locate the FILID entries for all system data files owned by the pseudo-user IDs (\*IMG, \*NLS, \*NSS, and \*UCR).

#### **\*UCR**

tells CP to locate the FILID entry for all user class restructure files owned by the system.

### Usage Notes

1. Pattern matching follows the rules used by the CMS LISTFILE command. Use an asterisk (\*) to match any number of characters; use a percent symbol (%) to match any single character. Because there are operands for pseudo-user ID files (\*IMG, \*NLS, \*NSS, \*UCR, and \*SDF) and because a single asterisk (\*) indicates your user ID, there are exceptions to the asterisk rule.

To specify a pattern that you would normally specify as one of these special cases, use an additional asterisk. For example, specify **\*\*NSS** to match all user IDs that end with the 3 characters NSS, including the pseudo-user ID **\*NSS**.

## Responses

Response 1:

```
OwnerID  FILID
ID       nnnnnnnn
:
```

### OwnerID

shows the user ID or pseudo-user ID.

### FILID

shows the host logical storage address of the spool file identification entry for that user ID or pseudo-user ID.

## Examples

Example 1:

When you enter a LOCATE FILID command with the **\*SDF** operand, you may see something like this:

```
locate filid *sdf
OwnerID  FILID
*IMG     0764D058
*NLS     0764D0A0
*NSS     0764D034
*UCR     0764D07C
Ready;
```

Example 2:

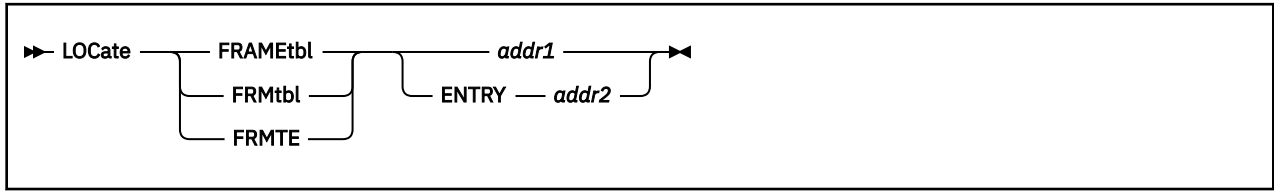
When you enter LOCATE FILID **\***, you may see this:

```
locate filid *
OwnerID  FILID
EMILY    0764D97C
Ready;
```

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP6720I No items found to satisfy selection criteria.

## LOCATE FRAMETBL



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE FRAMETBL, LOCATE FRMTBL, or LOCATE FRMTE to display the address of the frame table entry (FRMTE) for a specific storage frame, or the address of the frame for a specific FRMTE.

### Operands

#### **addr1**

is the host real storage address of the frame whose FRMTE you want CP to locate. This address need not be frame-aligned; CP will adjust it.

#### **ENTRY addr2**

tells CP to locate the frame whose FRMTE is located at *addr2*. Because FRMTEs reside above 2 GB in the system execution space, *addr2* must be a host logical storage address above 2 GB. This address need not be aligned with the boundary of a FRMTE; CP will adjust it.

### Responses

**Note:** If SET UNDERSCORE ON has been specified, an underscore is used to separate 16-digit addresses for readability.

Response 1:

If the frame address is < 4 GB, the response format is:

```
Frame Address FRMTE
nnnnnnnn      nnnnnnnnnnnnnnnnnnn
```

#### **Frame Address**

shows the host real storage address of the frame.

#### **FRMTE**

shows the host logical storage address of the frame table entry.

Response 2:

If the frame address is > 4 GB, the response format is:

```
Frame Address  FRMTE
nnnnnnnnnnnnnnnnnn  nnnnnnnnnnnnnnnnnnn
```

### Examples

Example 1:

To find the frame table entry for the frame at host real storage address 12f000, enter:

```
locate frmte 12f000
Frame Address FRMTE
0012F000      00000001390025E0
Ready;
```

The response shows that the FRMTE is located at host logical storage address 1390025E0.

Example 2:

To find the frame table entry for the frame at host real storage address 100000000, enter:

```
locate frmte 100000000
Frame Address FRMTE
0000000100000000 000000013B000000
Ready;
```

The response shows that the FRMTE is located at host logical storage address 13B000000.

Example 3:

To find the frame whose FRMTE is located at host logical storage address 139000100, enter:

```
locate frmte entry 139000100
Frame Address FRMTE
00008000      0000000139000100
Ready;
```

The response shows that the frame is located at host real storage address 8000.

Example 4:

To find the frame whose FRMTE is located at host logical storage address 13C100000, enter:

```
locate frmte entry 13c100000
Frame Address FRMTE
00000001_88000000 00000001_3C100000
Ready;
```

The response shows that the frame is located at host real storage address 188000000. Note that SET UNDERSCORE ON is in effect.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP004E Invalid hexloc - *addr*
- HCP025E Storage missing or invalid
- HCP026E Operand missing or invalid
- HCP094E Storage exceeds allowed maximum of *maximum* {M|K}
- HCP6762E Frame table entry is outside the extents of the frame table: *addr1-addr2*

## LOCATE ICLBK

```
▶▶ LOCate — ICLBK — epname ▶▶
```

### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE ICLBK to display the address of the CP indirect call locator block (ICLBK) for a specific entry point.

### Operands

#### *epname*

is the name of the entry point whose CP indirect call locator block address you want to display. The variable *epname* must be a 1-character to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the four special characters (\$, #, \_, and @), or any combination thereof.

You can use generic entry point names to request information about a specific subset of entry point names. A generic entry point name is a 1-character to 8-character string with asterisks (\*) in place of one or more characters and percent signs (%) in place of exactly one character. For example,

```
locate iclbk hcp%rc*
```

lists all the entry points that start with HCP and have RC as their fifth and sixth characters.

### Usage Notes

1. To load customer-written CP routines into the system execution space, use the CPXLOAD command (see [“CPXLOAD” on page 111](#)).
2. To add to, change, or replace the list of entry points and external symbols associated with an exit point and to enable or disable that exit point, use the ASSOCIATE EXIT command (see [“ASSOCIATE EXIT” on page 20](#)).
3. To display usage statistics information about indirect calls to external entry points, use the QUERY ICLNAME command (see [“QUERY ICLNAME” on page 943](#)).
4. For more information about the indirect call locator block, see [Available Diagnostic Facilities in z/VM: CP Exit Customization](#).

### Responses

Response 1:

```
Symbol   ICLBK
symbol   nnnnnnnn
:
```

#### Symbol

shows the entry point name you specified, or an entry point name that matches the pattern you specified.

**ICLBK**

shows the host logical storage address of the CP indirect call locator block for that entry point.

**Examples**

## Example 1:

To display the address of the indirect call locator block for entry point HCPSRC00, enter the following:

```
locate iclbnk hcpsrc00
Symbol   ICLBK
HCPSRC00 06FE09C8
Ready;
```

## Example 2:

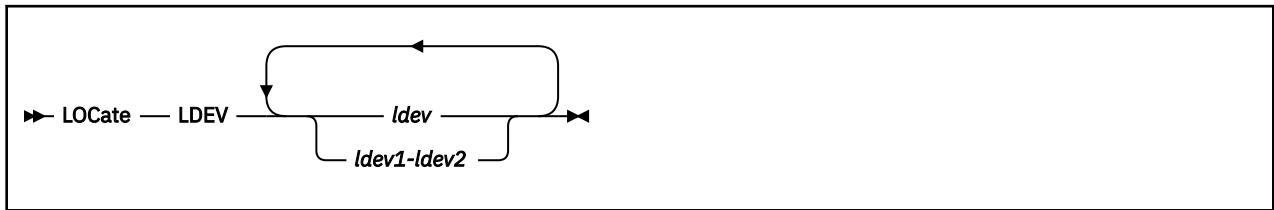
To display the address of all the indirect call locator blocks for entry points that have "HCPSRC" as the first six characters of their name, enter the following:

```
locate iclbnk hcpsrc*
Symbol   ICLBK
HCPSRC00 061E09C8
HCPSRC04 06F109C8
HCPSRC08 06FE01C8
HCPSRC10 06FE19C8
Ready;
```

**Messages**

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP6706E Invalid entry point name - *epname*
- HCP6720I No items found to satisfy selection criteria.

## LOCATE LDEV



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE LDEV to find the addresses of a system logical device block and its associated control blocks.

### Operands

*ldev*

*ldev1-ldev2*

is a logical device number, range of logical device numbers, or a combination of logical device numbers and ranges of logical device numbers.

### Responses

Response 1:

```
Dev  RDEV  LddbK  LSOBJ  Host
dev  nnnnnnn nnnnnnn nnnnnnn host
:
```

#### Dev

shows the logical device number.

#### RDEV

shows the host logical storage address of the corresponding real device block.

#### LddbK

shows the host logical storage address of the logical display device simulation block.

#### LSOBJ

shows the host logical storage address of the logical subchannel object.

#### Host

shows the user ID of the virtual machine that created the logical device.

### Examples

Example 1:

If you enter a LOCATE LDEV command for a range of device numbers, you may see something like this:

```
locate ldev 0000-0FFF
Dev  RDEV  LddbK  LSOBJ  Host
L0001 07AC71E8 07B80200 07AA31C8 TCPIP
L0006 07AC9788 07B80E00 07AA3C48 TCPIP
L0008 07AB95A8 07B80D40 07AA3A88 TCPIP
L000D 004A5008 07B80B60 01D4E008 TCPIP
L0010 004A53C8 07B80AA0 07AA3468 TCPIP
L0011 004A55A8 07B80B00 07AA38C8 TCPIP
```

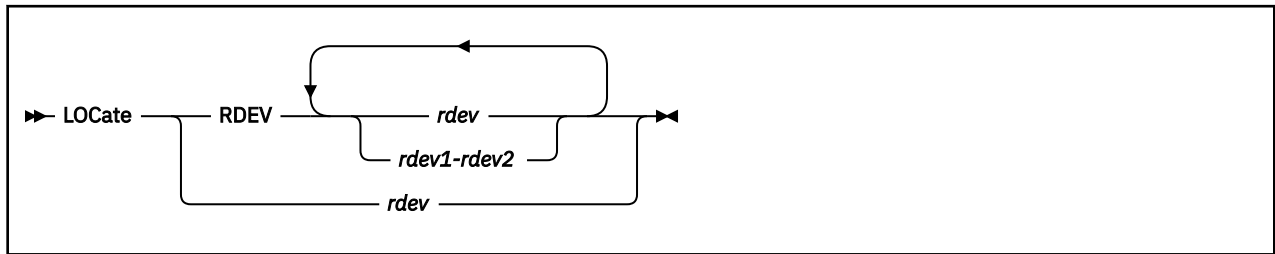


```
L0012 07AC95A8 07B80DA0 01D4E2A8 TCPIP  
L0014 004A5968 07B80A40 01D4E8C8 TCPIP  
L0015 004A5B48 07B80980 01D4E7E8 PVM  
L001A 0188A008 07B80E60 01D4E388 TCPIP  
Ready;
```

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP009E Invalid range - *range*
- HCP026E Operand missing or invalid
- HCP1100E A logical device number was not supplied or it was invalid.
- HCP6720I No items found to satisfy selection criteria

## LOCATE RDEV



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE RDEV to display the addresses of a real device block and its associated control blocks.

### Operands

*rdev*

*rdev1-rdev2*

specifies a real device number, a range of real device numbers, or a combination of real device numbers and ranges of real device numbers displays the address of the real device block (RDEV) and, if there are any, the addresses of its associated control blocks. For a device in the active configuration, the device number entered must be a 4-digit hexadecimal number between X'0000' and X'FFFF'. For a device in the standby configuration, the device number must be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured. A range of devices cannot span subchannel sets.

### Usage Notes

1. Because there are several LOCATE commands that treat the qualifying keyword as optional (LOCATE RDEV, LOCATE SYMBOL, LOCATE VDEV, LOCATE VMDBK), specifying the RDEV keyword is recommended to avoid undesired results. Using RDEV also allows you to specify more than one real device number.
2. LOCATE *rdev* is not a valid command for SNA/CCS terminals, which are connected through a VTAM service machine and have no real device address known to CP. However, each logged-on or dialed SNA/CCS terminal has a dynamic RDEV control block. For SNA/CCS terminals, the user finds the RDEV control block by using the LOCATE VDEV *userid vdev* command (see “LOCATE VDEV” on page 639). This command returns the address of the VDEV control block, and the VDEV contains a pointer to the RDEV.

### Responses

Response 1:

```

Dev  RDEV    CPVOL    VEXBK    RSPBK    SPFBK
dev  hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
:
```

#### Dev

shows the real device number.

#### RDEV

shows the host logical storage address of the corresponding real device block.

**CPVOL**

shows the host logical storage address of the system volume list entry.

**VEXBK**

shows the host logical storage address of the volume extent block, which contains information on all the extents on a CP-owned volume.

**RSPBK**

shows the host logical storage address of the real spool device block.

**SPFBK**

shows the host logical storage address of the spool file control block.

**Note:**

1. If a single device is specified on the command, the Dev field is omitted from the response.
2. An asterisk (\*) in a response field indicates there is no information for that field.
3. If there is no information for a response field for all of the devices listed in the response, the column is omitted.

**Examples****Example 1:**

If you enter a LOCATE RDEV command for multiple devices, you may see something like this:

```
locate rdev 0500-0504 30100 0a01-0a02 0e0e
Dev RDEV CPVOL VEXBK RSPBK SPFBK
0500 07AFB3C8 * * * *
0501 07AFB5A8 * * * *
0502 07AFB788 * * * *
0503 07AFB968 * * * *
0504 07AFBB48 * * * *
30100 07B1A3C8 * * * *
0A01 07AF05A8 07B1CB30 07B6602C * *
0A02 07AF0788 07B1CB68 07B6B02C * *
0E0E 07ADC1E8 * * 07AA30E8 075ECD28
Ready;
```

**Example 2:**

```
locate rdev 30100
RDEV
07B1A3C8
Ready;
```

This example shows that there are no control blocks associated with this real device block from a device in the standby configuration.

**Example 3:**

```
locate rdev 0a01
RDEV CPVOL VEXBK
07AF05A8 07B1CB30 07B6A02C
Ready;
```

This example shows that the CPVOL and VEXBK control blocks are associated with this real device block, and the response gives their addresses.

**Example 4:**

```
locate rdev 0e0e
RDEV RSPBK SPFBK
07ADC1E8 07AA40E8 0760BB68
Ready;
```

This example shows that the RSPBK and SPFBK control blocks are associated with this real device block, and the response gives their addresses.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP009E Invalid range - *range*
- HCP021E A real device number was not specified or it is invalid.
- HCP026E Operand missing or invalid
- HCP6720I No items found to satisfy selection criteria.

## LOCATE RPCI



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE RPCI to display the address of a real PCI function block and its associated control blocks.

### Operands

**rpfid**

**rpfid1-rpfid2**

specifies a real PCI function ID or a range of PCI function IDs whose control block addresses are to be displayed. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. The total number of PCI functions specified cannot exceed 256.

### Responses

Response 1:

```
PCIF      RPCI      VPCI
nnnnnnnn nnnnnnnn -----
```

Response 2:

```
PCIF      RPCI      VPCI
nnnnnnnn nnnnnnnn nnnnnnnn
```

Where:

#### PCIF

shows the real PCI function ID

#### RPCI

shows the host logical storage address of the corresponding real PCI block.

#### VPCI

shows the virtual PCI function block address if the PCI function is attached to a user ID.

### Examples

Example 1:

If you enter a LOCATE RPCI 18 command, you may see something like this:

```
PCIF      RPCI      VPCI
00000018 02E36288 -----
```

Example 2:

## LOCATE RPCI

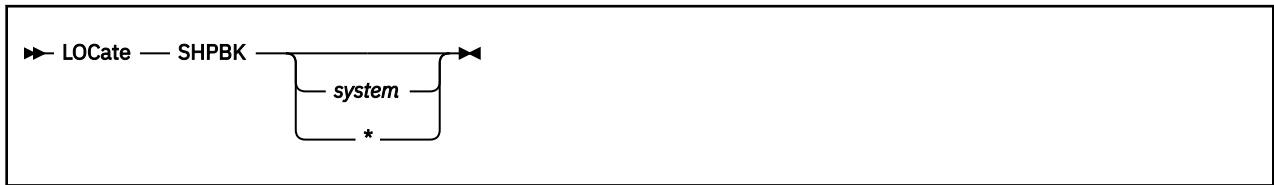
If you enter a LOCATE RPCI 18 command and PCIF 18 has been attached to a user, you may see something like this:

```
PCIF      RPCI      VPCI
00000018 02E36288 012E8708
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP021E A real PCI function ID was not supplied or it is invalid.
- HCP5780E Adjunct does not support the *command subcommand* command
- HCP6000E The range of PCI function ids cannot exceed 256.
- HCP6720I No items found to satisfy selection criteria

## LOCATE SHPBK



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE SHPBK to display the address of the shared system name table control block (SHPBK).

### Operands

#### *system*

tells CP to locate the control blocks for only those system names that match this pattern. For more information on system patterns, see Usage Note “1” on page 623.

If the system names do not match, the system alias is also checked for a match.

If you do not specify a system name, CP locates all SHPBKs.

\*

tells CP to locate the SHPBK for the current system.

### Usage Notes

1. Pattern matching follows the rules used by the CMS LISTFILE command. Use an asterisk (\*) to match any number of characters; use a percent symbol (%) to match any single character.

### Responses

Response 1:

```

Slot  System  SHPBK   ISFSYSTB
   n  system  nnnnnnnn nnnnnnnn [RESERVED]
:
  
```

#### **System**

shows the system name.

#### **SHPBK**

shows the host logical storage address of the shared system name table control block for that system.

#### **ISFSYSTB**

shows the host logical storage address of the CSE/SSI system table.

#### **RESERVED**

indicates there is a "reserved" system. In an SSI cluster, this indicates an available slot in the SSI member list.

### Examples

Example 1:

## LOCATE SHPBK

If you enter a LOCATE SHPBK command with no other operands, you may see something like this:

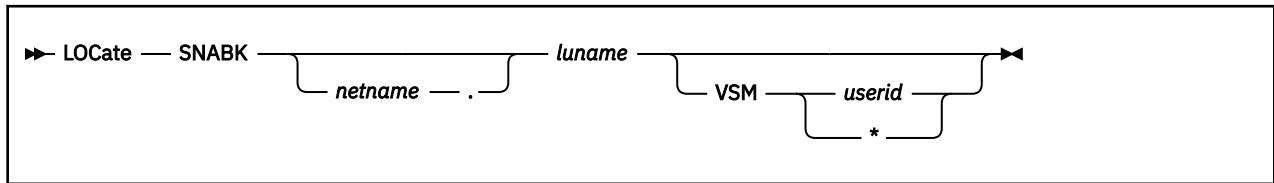
```
locate shpbk
Slot System  SHPBK    ISFSYSTB
  1  -----  0001CAF8 0001CEB0 RESERVED
  2  BOSTON2  0001CBF8 0001CEB0
  3  BOSTON3  0001CB78 0001CEB0
Ready;
```

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP6720I No items found to satisfy selection criteria.



## LOCATE SNABK



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE SNABK to display the address of the system SNA control block (SNABK).

### Operands

#### *netname.*

is the network qualifier of the fully qualified logical unit name (*netname.luname*). The period is used to delimit the network name from the LU name. This tells CP to locate the control blocks for only those terminal names that match this pattern.

#### *luname*

tells CP to locate the control blocks for only those terminal names that match this pattern.

#### VSM

tells CP to display the addresses for those SNA control blocks under the control of any VTAM Service Machine whose user ID matches the pattern *userid*.

#### *userid*

tells CP to locate the SNABK entries for the user ID or user ID pattern specified. For more information on user ID patterns, see Usage Note [“1” on page 625](#).

#### \*

tells CP to locate the SNABK entries for your user ID.

### Usage Notes

1. Pattern matching follows the rules used by the CMS LISTFILE command. Use an asterisk (\*) to match any number of characters; use a percent symbol (%) to match any single character.
2. Blanks are not permitted between the *netname*, ., and *luname*.
3. Network qualifiers, if they exist will be displayed only if a *netname* is specified.
4. Multiple responses can occur with the same *luname* when a *netname* is not specified and one or more VSMs support network qualified names.

### Responses

Response 1:

```
LU Name  RDEV    SNABK    VSM
luname  nnnnnnnn nnnnnnnn userid
:
```

#### LU Name

shows the name of the logical unit.

## LOCATE SNABK

### RDEV

shows the host logical storage address of the real device block.

### SNABK

shows the host logical storage address of the SNA control block.

### VSM

shows the user ID of the VTAM service machine (VSM) controlling the SNABK.

## Examples

Example 1:

If you enter a LOCATE SNABK command without the VSM operand, you may see something like this:

```
locate snabk wgu*
LU Name  RDEV      SNABK      VSM
WGU00315 044D9060 044D9008  VTAM
Ready;
```

Example 2:

If you add the *netname* to the request the response shows the fully qualified network name and might look like this:

```
locate snabk *.*
LU Name  RDEV      SNABK      VSM
VTAM1.LU3E0 0667DCD0 0667DC68  VTAM
VTAM1.LU3E1 07077D88 07077D20  VTAM
Ready;
```

The same request without the *netname* might look like this:

```
locate snabk *
LU Name  RDEV      SNABK      VSM
LU3E0    0667DCD0 0667DC68  VTAM
LU3E1    07077D88 07077D20  VTAM
Ready;
```

Example 3:

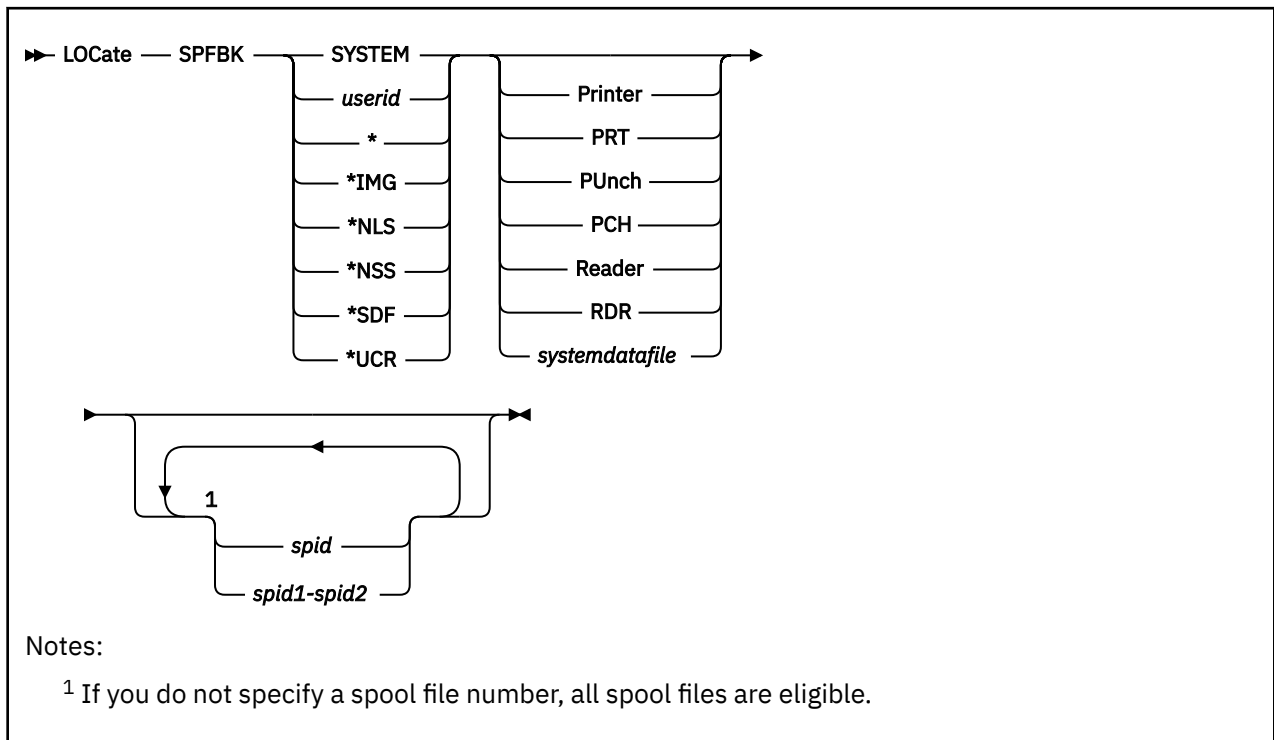
You can request the addresses of the SNA control blocks for all the logical devices under the control of a specific VSM. For example, to show the relevant information for all the terminals whose names start with the characters TN6 that are logged on under VTAM2, enter:

```
locate snabk tn6* vsm vtam*
LU Name  RDEV      SNABK      VSM
TN6PV095 044D92A0 044D9248  VTAM2
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP6720I No items found to satisfy selection criteria

## LOCATE SPFBK



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE SPFBK to find the addresses of the spool file control blocks (SPFBKs) for specified user IDs, pseudo-user IDs, or user ID patterns.

### Operands

#### *userid*

tells CP to locate the SPFBK entries for the user ID, user ID pattern, or pseudo-user ID (`*IMG`, `*NLS`, `*NSS`, `*UCR`, and `*SDF`) specified. For more information on user ID patterns, see Usage Note “1” on page 628.

#### **SYSTEM**

tells CP to locate the SPFBK entries for all user IDs. In this case, any spool file numbers are system spool file numbers, not user spool file numbers.

**\***

tells CP to locate the SPFBK entry for your user ID.

#### **\*IMG**

tells CP to locate the SPFBK entry for all printer image library files owned by the system.

#### **\*NLS**

tells CP to locate the SPFBK entry for all National Language Support files owned by the system.

#### **\*NSS**

tells CP to locate the SPFBK entry for all named saved system files owned by the system.

## LOCATE SPFBK

### **\*SDF**

tells CP to locate the SPFBK entries for all the system data files owned by the pseudo-user IDs (\*IMG, \*NLS, \*NSS, and \*UCR).

### **\*UCR**

tells CP to locate the SPFBK entry for all user class restructure files owned by the system.

### **Printer**

#### **PRT**

specifies printer spool files.

### **PUnch**

#### **PCH**

specifies punch spool files.

### **Reader**

#### **RDR**

specifies reader spool files.

### **systemdatafile**

specifies a system data file. System data files can be any of the following: IMG, NLS, NSS, TRF or UCR. SDF means all of these. If you do not specify a spool file type, all spool file types are eligible.

### **spid**

#### **spid1-spid2**

specifies a spool file number, range of spool file numbers, or a combination of spool file numbers and range of spool file numbers for the spool files desired. If you do not specify a spool file number, all spool files are eligible.

## Usage Notes

1. Pattern matching follows the rules used by the CMS LISTFILE command. Use an asterisk (\*) to match any number of characters; use a percent symbol (%) to match any single character. Because there are operands for pseudo-user IDs (\*IMG, \*NLS, \*NSS, \*UCR, and \*SDF) and because a single asterisk (\*) indicates your user ID, there are exceptions to the asterisk rule.

To specify a pattern that you would normally specify as one of these special cases, use an additional asterisk. For example, specify \*\*NSS to match all user IDs that end with the 3 characters NSS, including the pseudo-user ID \*NSS.

## Responses

Response 1:

```
OwnerID  SpID Type SPFBK   System  System-SpID
ownerid  spid type nnnnnnnn system  system_spid
:
```

### **OwnerID**

shows the user ID that owns the spool file.

### **SpID**

shows the spool file identification number.

### **Type**

shows the type of spool file.

### **SPFBK**

shows the host logical storage address of the spool file control block.

### **System**

shows the name of the system that controls the spool file.

### **System-SpID**

shows the system spool file identification number for the specified file. This number uniquely identifies each spool file on the system.

## Examples

Example 1:

If you enter LOCATE SPFBK with the SYSTEM operand, you may see something like this:

```
locate spfbk system
OwnerID SpID Type SPFBK System System-SpID
OLIVER 7369 RDR 0763A1C8 BOSTON 31260
DAVID 6951 RDR 07647A88 BOSTON 45996
CORAK 2179 RDR 075FC468 BOSTON 2049
TERRI 0042 RDR 076097E8 BOSTON 8149
DEGENARO 0395 RDR 076310E8 BOSTON 26096
MURPHY 0406 RDR 07B60AB8 BOSTON 205134
PIERCE 3018 RDR 07647D28 BOSTON 44773
HEATHER 0052 RDR 07B61D58 BOSTON 149558
DONNA 0033 RDR 07630E08 BOSTON 24805
TERRI 0055 RDR 076279A8 BOSTON 19589
EMILY 0242 RDR 0762DEE8 BOSTON 26419
OPERATOR 1675 PRT 07627D28 BOSTON 19538
*IMG 0001 IMG 0763B388 BOSTON 32055
*NSS 0016 NSS 075F40E8 BOSTON 20
Ready;
```

Example 2:

If you use the \*SDF operand, you may see this:

```
locate spfbk *sdf
OwnerID SpID Type SPFBK System System-SpID
*IMG 0001 IMG 0763B388 BOSTON 32055
*NSS 0064 NSS 07647E08 BOSTON 44739
*NSS 0090 NSS 075FDA88 BOSTON 2524
*NSS 0055 NSS 07B6CAB8 BOSTON 97316
Ready;
```

Example 3:

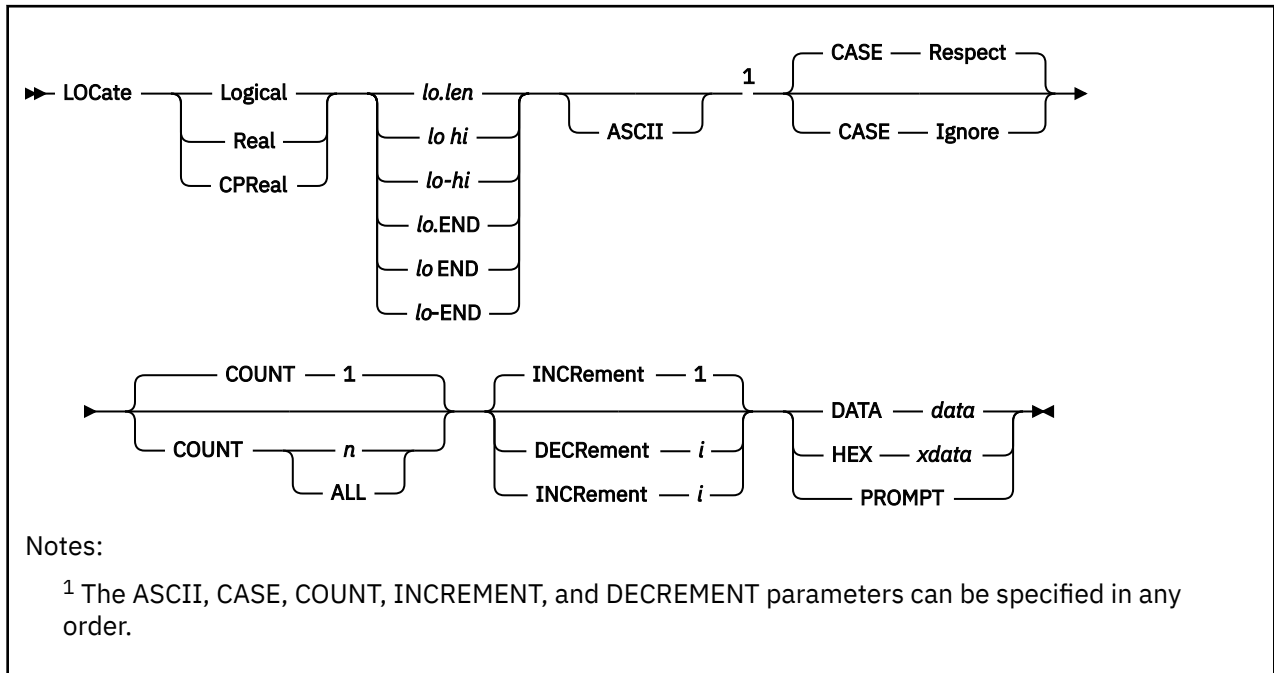
If you use the asterisk (\*) operand, you may see this:

```
locate spfbk *
OwnerID SpID Type SPFBK System System-SpID
EMILY 0242 RDR 0762DEE8 BOSTON 26419
EMILY 0257 RDR 0763E0E8 BOSTON 27056
EMILY 0260 PRT 0424DD08 BOSTON 27462
Ready;
```

## Messages

- HCP007E Invalid userid - *userid*
- HCP008E Invalid spoolid - *spoolid*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP6720I No items found to satisfy selection criteria

## LOCATE (Storage)



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE (Storage) to display the location of data in host logical storage or host real storage.

### Operands

#### Logical

searches host logical storage (the system execution space).

#### Real

#### CPReal

searches host real storage.

#### *lo.len*

specifies the address of the beginning of the search area and the number of bytes to its end. *lo* and *len* must both be hexadecimal numbers. You must enter a period between these two values.

#### *lo hi*

#### *lo-hi*

specifies the addresses for the beginning and end of the search area. *lo* and *hi* must both be hexadecimal.

*lo*, *len*, and *hi* can be a 1 to 16 hexadecimal digit 64-bit address, separated with an optional underscore for readability.

#### *lo.END*

#### *lo END*

#### *lo-ND*

specifies the address of the beginning of the search area and tells CP to continue to search through to the end of the available storage. *lo* is a hexadecimal number. You can use any of the above formats.

**ASCII**

tells CP to convert the data you enter to ASCII before it is compared to the data in storage.

**CASE**

tells CP to convert the data you enter to upper case before it compares it to the data in storage.

**Note:** Unless you are using DIAGNOSE code X'08' to send commands to CP in mixed case, CP converts commands entered from the terminal to upper case before it processes them. If you specify the HEX operand, CASE is ignored. The default is CASE RESPECT.

**Respect**

tells CP not to convert the data from the command and the data from storage before comparing them.

**Ignore**

tells CP to convert the data from the command and the data from storage before comparing them so the upper and lower case characters are considered equal.

**COUNT *n*****COUNT ALL**

specifies how many matches to report. ALL reports all matches. If COUNT is not specified, the default is COUNT 1.

**DECRement *i***

tells CP to go *i* number of bytes backward through storage to search for the data. *i* must be a hexadecimal number. CP begins at the address you have specified as the end of the search area and compares the data you enter with what is at that address. If it does not find a match, it goes back *i* bytes and tries again. When it reaches the beginning of the search area or finds as many matches as you have specified with the COUNT keyword, CP displays the addresses of the fields within the search area that contain the data you are looking for.

*i* can be a 1 to 16 hexadecimal digit 64-bit address, separated with an optional underscore for readability. If the underscore is used there must be 8 hexadecimal digits to the right of it.

**INCRement *i***

tells CP to go *i* number of bytes forward through storage to search for the data. *i* must be a hexadecimal number. CP begins at the address you have specified as the start of the search area and compares the data you enter with what is at that address. If it does not find a match, it advances *i* bytes and tries again. When it reaches the end of the search area or finds as many matches as you have specified with the COUNT keyword, CP displays the addresses of the fields within the search area that contain the data you are looking for.

If you do not specify INCREMENT or DECREMENT, INCREMENT 1 is assumed.

*i* can be a 1 to 16 hexadecimal digit 64-bit address, separated with an optional underscore for readability. If the underscore is used there must be 8 hexadecimal digits to the right of it.

**DATA *data***

specifies the data to search for. The data to be compared starts after the first blank following the keyword DATA. Leading, imbedded, and trailing blanks are included in the comparison.

**HEX *xdata***

specifies the hexadecimal data to search for. The data may be specified as any number of strings of hexadecimal digits. If necessary, each string will be padded on the left with a single leading zero (0) to generate a string of an even number of hexadecimal digits. CP ignores blanks between the hexadecimal strings as it converts the strings for comparison. The length of the data to search for must be less than 254 hexadecimal digits.

**PROMPT**

tells CP to prompt you to enter the data to be searched for. The length of the data to search for must be less than 255 bytes.

### Usage Notes

1. CP searches only addresses for pages assigned. CP skips a page for an address not assigned (for example, an unreferenced page). If CP skips a page for any reason, the next comparison is in the next acceptable page at the address that would have been reached had the page or pages not been skipped and had all comparisons failed.
2. If you specify the search area as *lo hi* or *lo-hi*, CP searches and compares the byte at address *hi*. For example,

```
locate logical 0 1 hex 00
```

causes CP to compare the two bytes (address 0 and address 1) to the specified data (00).

3. Specifying the search area as *lo hi* or *lo-hi* is the same as specifying *lo.len* if  $len = hi - lo + 1$ .
4. If CP finds a match, the address of the next comparison attempt is the address of the match plus the increment amount (or minus the decrement amount).
5. If you specify DECREMENT, CP displays the addresses of matches with the highest address first.
6. Whether you specify INCREMENT or DECREMENT, the set of addresses eligible for the search are the same. The difference is that INCREMENT causes the search to start from the *lo* address and progress upward, while DECREMENT causes the search to start at the *hi* address and progress downward ending at the *lo* address. For example, in the following command:

```
locate logical 0 1000 decr c data 123456
```

the first comparison address is X'FF0'.

7. CP skips pages that contain CP directory data.
8. If a large search range is specified, the LOCATE command will take a large amount of CPU cycles and therefore can impact overall system performance.

### Responses

Response 1:

```
The data is located at address.  
:
```

**Note:** A host logical storage address is 8 hexadecimal digits. A host real storage address can be up to 16 hexadecimal digits, separated by an underscore if SET UNDERSCORE ON has been specified.

Response 2:

If you specify the PROMPT operand, you see the following:

```
Enter data now. To abort the command,  
press ENTER with no data or press PA1.
```

### Examples

Example 1:

If you enter a LOCATE REAL command, you might see something like this:

```
locate real 1_00000000-2_00000000 data david  
The data is located at 1_0001B8C8.  
Ready;
```

Example 2:

The following commands are equivalent:



```
locate real 0-400 hex 00010203
The data is located at 00000045.
Ready;
```

```
locate cpreal 0-400 hex 0 1 203
The data is located at 00000045.
Ready;
```

### Example 3:

If you enter a LOCATE REAL command with the COUNT operand for data you know is in the CP nucleus, you might see something like this:

```
locate real 2e8000-2eb000 count 3 data hcp
The data is located at 002E800C.
The data is located at 002E900C.
The data is located at 002EA00C.
Ready;
```

### Example 4:

If you enter a LOCATE LOGICAL command for the same data as in Example 3, you might see something like this:

```
locate logical 2e8000-2eb000 count 3 data hcp
The data is located at 002E800C.
The data is located at 002E900C.
The data is located at 002EA00C.
Ready;
```

Because the CP nucleus is identity mapped, the host logical storage addresses of the data are identical to the host real storage addresses. For data outside the CP nucleus, the host real storage addresses and host logical storage addresses would be different.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP004E Invalid hexloc - *hexloc*
- HCP005E Invalid hexdata - *data*
- HCP009E Invalid range - *range*An invalid address range was specified
- HCP026E Operand missing or invalid
- HCP6704E Missing token at end of line
- HCP6720I No items found to satisfy selection criteria
- HCP6896E The data field exceeds the maximum length of *nnn* characters.

## LOCATE SXSTE



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE SXSTE to display the address of the system execution space page management table entry (SXSTE) for a specific system execution space page, or the address of the system execution space page for a specific SXSTE.

### Operands

#### *addr1*

is the host logical storage address of the system execution space page whose SXSTE you want CP to locate. This address does not need to be page-aligned; CP will adjust it.

#### **ENTRY** *addr2*

tells CP to locate the system execution space page whose SXSTE address is *addr2*. Because SXSTEs reside above 2 GB in the system execution space, *addr2* must be a host logical storage address above 2 GB. This address does not need to be aligned with the boundary of a SXSTE; CP will adjust it.

### Responses

**Note:** If SET UNDERSCORE ON has been specified, an underscore is used to separate 16-digit addresses for readability.

Response 1:

```
Page Address SXSTE
nnnnnnnn  nnnnnnnnnnnnnnnnn
```

#### **Page Address**

shows the host logical storage address of the system execution space page.

#### **SXSTE**

shows the host logical storage address of the system execution space page management table entry.

### Examples

Example 1:

To find the SXSTE address for page 19FE400, enter:

```
locate sxste 19FE400
Page Address SXSTE
019FE000    0000020028033FC0
Ready;
```

The response shows that the SXSTE is located at host logical storage address 20028033FC0.

Example 2:

To find the page whose SXSTE is located at host logical storage address 2003E011020, enter:

```
locate sxste entry 2003E011020
Page Address SXSTE
00881000      00000200_3E011020
Ready;
```

The response shows that the page is located at host logical storage address 881000. Note that SET UNDERSCORE ON is in effect.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP004E Invalid hexloc - *hexloc*
- HCP025E Storage missing or invalid
- HCP094E Storage exceeds allowed maximum of *maximum* {M|K}
- HCP6875E System execution space page management table entry is outside the extents of the SXSTE table: *addr1-addr2*

## LOCATE SYMBOL



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE SYMBOL to display the address of an external symbol.

### Operands

#### *symbol*

is the name of an external symbol. If you specify the SYMBOL keyword, CP supports pattern matching as described below in Usage Note “2” on page 636. Specifying a single asterisk (\*) with the SYMBOL keyword tells CP to list every CP external symbol and its location.

### Usage Notes

1. Because there are several LOCATE commands that treat the qualifying keyword as optional (LOCATE RDEV, LOCATE SYMBOL, LOCATE VDEV, LOCATE VMDBK), specifying the SYMBOL keyword is recommended to avoid undesired results. Specifying the SYMBOL keyword also allows you to use pattern matching in the symbol name, as described in Usage Note “2” on page 636, and provides additional information in the response.
2. Pattern matching generally follows the rules used by the CMS LISTFILE command. Use an asterisk (\*) to match any number of characters; use a percent symbol (%) to match any single character.

### Responses

Response 1:

If you specify the SYMBOL keyword, the response format is:

```
symbol = nnnnnnnn Resident nnnnnn Bytes
:
```

#### *symbol*

is the symbol name you specified, or a symbol name that matches the pattern you specified.

#### *nnnnnnnn Resident*

is the host logical storage address of the external symbol. All external symbols are resident.

#### *nnnnnn Bytes*

is the length of the CSECT, and is shown if the entry in the map contains a nonzero length for the external label:

- For normal assembler routines, the CSECT is longer than 0 bytes. Therefore, when you enter a LOCATE command for an assembler module, CP shows you the length of the entire module. If you enter a LOCATE command for an entry point within an assembler module, CP does not display any information about the module's length, because entry points are not defined to have length.

- For PLX routines, the first CSECT is 0 bytes long. If you enter a LOCATE command for a PLX module, CP does not display any information about the length of the entire module. Entry points within a PLX module are actually independent CSECTs and are longer than 0 bytes. When you enter a LOCATE command for an entry point within a PLX module, CP shows you the length of the CSECT indicated by that entry point.

See Examples 2, 3, and 4.

Response 2:

If you do not specify the SYMBOL keyword, the response format is:

```
symbol = nnnnnnnn Resident
```

## Examples

Example 1:

If you use LOCATE SYMBOL without the SYMBOL keyword, you may see the following:

```
locate hcpabc
HCPABC = 00868000 Resident
Ready;
```

This response indicates HCPABC is at host logical storage address 00868000.

Example 2:

If you use LOCATE SYMBOL with the SYMBOL keyword to find the address of the same external symbol, you may get the following response:

```
locate symbol hcpabc
HCPABC = 00868000 Resident      000858 Bytes
Ready;
```

Example 3:

If you use pattern matching to find the addresses of all the external symbols that start with the letters HCPZSP, you may see the following:

```
locate symbol hcpzsp*
HCPZSP = 003D1CB0 Resident      0000A8 Bytes
HCPZSPPP = 003D1D30 Resident
HCPZSPRD = 003D1D10 Resident
Ready;
```

This response shows that HCPZSP is an Assembler module, and the other two items listed are entry points within it.

Example 4:

If you use pattern matching to find the addresses of all the external symbols that start with the letters HCPCFC, you may see the following:

```
locate symbol hcpcfc*
HCPCFC = 0087BAD0 Resident
HCPCFCMD = 0087BAD0 Resident      000900 Bytes
HCPCFCSD = 0087C3D0 Resident      0003A8 Bytes
Ready;
```

This response shows that HCPCFC is a PLX module, and the other two items are entry points within it.

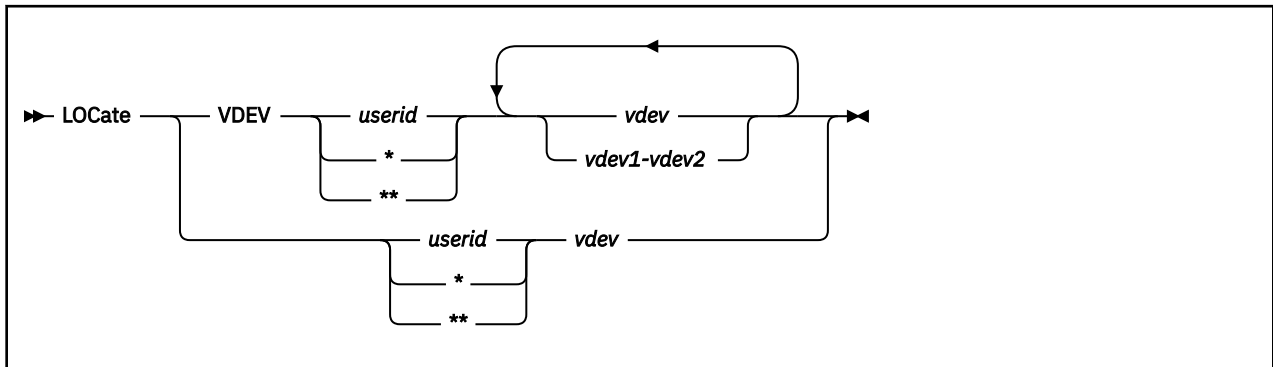
## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*

## LOCATE SYMBOL

- HCP026E Operand missing or invalid
- HCP1003E *External symbol name* {is|was} not found.
- HCP6720I No items found to satisfy selection criteria

## LOCATE VDEV



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE VDEV to display the addresses of the CP control blocks associated with a user or a user's virtual device.

### Operands

#### *userid*

\*

identifies the virtual machine whose control blocks you want to locate. To display the addresses of control blocks associated with your own virtual machine, specify *userid* as an asterisk (\*).

\*\*

indicates this is for all users logged onto the system.

#### *vdev*

#### *vdev1-vdev2*

identifies a virtual device number or range of virtual device numbers.

### Usage Notes

1. Because there are several LOCATE commands that treat the qualifying keyword as optional (LOCATE RDEV, LOCATE SYMBOL, LOCATE VDEV, LOCATE VMDBK), specifying the VDEV keyword is recommended to avoid undesired results. Using the VDEV keyword also allows you to specify more than one virtual device number and returns more information in the response.
2. You can use the LOCATE VDEV *userid vdev* command to find the RDEV control block for SNA/CCS terminals. This command returns the address of the VDEV control block, which contains a pointer to the RDEV.

### Responses

Response 1:

If you specify the VDEV keyword, the response format is:

```
Userid  Dev  VDEV   VDSBK   VPXBK   VSPBK   Rdev
userid  nnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnn
:
```

## LOCATE VDEV

### Userid

shows the user ID you specified.

### Dev

shows the virtual device number you specified, or a device number included in a range you specified.

### VDEV

shows the host logical storage address of the virtual device block associated with that device number.

### VDSBK

shows the host logical storage address of the virtual device simulation block associated with that device number, if the device is a spooled device.

### VPXBK

shows the host logical storage address of the virtual printer extension block.

### VSPBK

shows the host logical storage address of the virtual spooling device block.

### Rdev

shows the real device number of the device you specified.

**Note:** An asterisk in a response field indicates there is no information for that field.

Response 2:

If you do not specify the VDEV keyword, the response format is:

```
VMDBK      VDEV
nnnnnnnn  nnnnnnnn
```

### VMDBK

shows the host logical storage address of the virtual machine definition block.

### VDEV

shows the host logical storage address of the virtual device block.

## Examples

Example 1:

If you use the LOCATE VDEV command to display the addresses of control blocks for a range of virtual devices for the DAVID user ID, you may see something like this:

```
locate vdev david 0100-0150
Userid  Dev  VDEV  VDSBK  VPXBK  VSPBK  Rdev
DAVID   0120 04829998 *      *      *      0287
DAVID   0121 04829A70 *      *      *      0288
DAVID   0122 04829E28 *      *      *      0285
DAVID   0123 04829F00 *      *      *      0284
DAVID   0124 044D3008 *      *      *      0289
DAVID   0125 044D30E0 *      *      *      0283
DAVID   0126 044D31B8 *      *      *      0284
Ready;
```

Example 2:

If you enter the LOCATE LDEV command to display all user IDs (using the pattern \*\*) that have a virtual device defined at 0009 (the typical virtual device for the virtual machine console), you may see something like this:

```
locate vdev ** 9
Userid  Dev  VDEV  VDSBK  VPXBK  VSPBK  Rdev
CORAK   0009 01FD7B10 018B1F48 *      02A2AF38 TN6PV013
OPERATOR 0009 06E6C008 01AA8F48 *      06E6C0E0 *
DAWN    0009 072F8590 01EF6F48 *      072F8668 WGU00315
RSCS6   0009 0201E560 02030F48 *      0201E638 *
RSCS5   0009 022D0858 02024F48 *      022D0930 *
VTAM    0009 027974F0 0260BF48 *      027975C8 *
MURPHY  0009 02A1B1B8 028CBF48 *      02A1B290 L0002
```



```

OLIVER  0009 0497CBA8 04979F48 *      0497CC80 L0000
DAVID   0009 04981478 0497FF48 *      04981550 L0001
PUBSAUTH 0009 04982A28 04984F48 *      04982B00 *
SVCDIR  0009 056BAAC0 04A6EF48 *      056BAB98 08E0

```

Example 3:

If you do not use the VDEV keyword, you may see something like this:

```

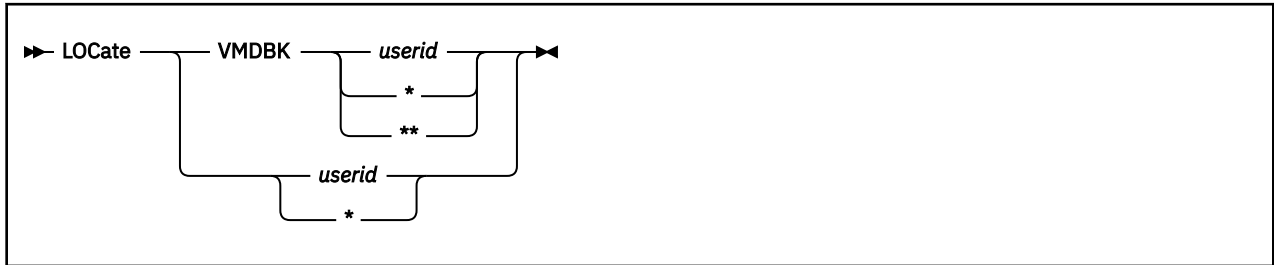
locate david 120
VMDBK      VDEV
04A78000   04829998
Ready;

```

## Messages

- HCP003E Invalid option - *option*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP6720I No items found to satisfy selection criteria

## LOCATE VMDBK



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE VMDBK to display the logical and real addresses of a user's virtual machine definition block (VMDBK).

### Operands

#### *userid*

specifies the user ID (or user ID pattern) whose VMDBK entries CP should locate. If you use the VMDBK keyword, CP supports pattern matching as described below in Usage Note “2” on page 642.

\*

tells CP to locate the VMDBK entry for your user ID.

\*\*

tells CP to locate all VMDBKs on the system.

### Usage Notes

1. Because there are several LOCATE commands that treat the qualifying keyword as optional (LOCATE RDEV, LOCATE SYMBOL, LOCATE VDEV, LOCATE VMDBK), specifying the VMDBK keyword is recommended to avoid undesired results. It also allows you to use pattern matching in specifying the user ID, as described in Usage Note “2” on page 642, and to see information about multiple virtual processors in the response.
2. Pattern matching follows the rules used by the CMS LISTFILE command. Use an asterisk (\*) to match any number of characters; use a percent symbol (%) to match any single character.
3. If a large virtual machine is logged off multiple times and two or more logged-off instances are still going through memory reset, the user ID (transformed from uppercase to lowercase) will appear multiple times in the responses to this command. Eventually, each instance will finish memory reset and will disappear from the responses.

The following sequence of events is an example of this type of situation:

- a. A user logs on virtual machine "LARGEONE" and runs a workload that instantiates a large amount of memory.
- b. The user logs off the virtual machine. This instance of virtual machine "LARGEONE" becomes known as instance "largeone."
- c. Responses to LOCATE VMDBK include "largeone nnnnnnnn LOGOFF nnnnnnnnnnnnnnnnn."
- d. The user logs back on to virtual machine "LARGEONE" and runs a workload that instantiates a large amount of memory.

- e. The user logs off. This instance of virtual machine "LARGEONE" also becomes known as instance "largeone."
- f. Responses to LOCATE VMDBK now include two occurrences of "largeone nnnnnnnn LOGOFF nnnnnnnnnnnnnnnn."
- g. Eventually instance "largeone" from Step "3.b" on page 642 completes reset and disappears.
- h. Responses to LOCATE VMDBK now include only one occurrence of "largeone nnnnnnnn LOGOFF nnnnnnnnnnnnnnnn."
- i. Eventually instance "largeone" from Step "3.e" on page 643 completes reset and disappears.

## Responses

**Note:** If SET UNDERSCORE ON has been specified, an underscore is used to separate 16-digit addresses for readability.

Response 1:

If you specify the VMDBK keyword, the response format is:

```
Userid    VMDBK    Type      VMDBK-Real
userid    nnnnnnnn type      nnnnnnnnnnnnnnnn
:
```

### Userid

shows the user ID you specified, or a user ID that matches the pattern you specified.

### VMDBK

shows the host logical memory address of the virtual machine definition block.

### Type

shows the type of VMDBK. The possible values are:

#### SYSTEM

is a special global VMDBK for CP.

#### ORIGIN

is a VMDBK associated with a user when that user logs on.

#### MP

is a VMDBK for an additional virtual processor defined by a user or the user's directory entry.

#### PROTOTYPE

is a VMDBK associated with a user in the process of logging off.

#### LOGOFF

indicates that the user has logged off but its memory is still in the process of being released.

### VMDBK-Real

shows the host real memory address of the virtual machine definition block.

Response 2:

If you do not specify the VMDBK keyword, the response format is:

```
VMDBK = nnnnnnnn VMDBK-Real = nnnnnnnnnnnnnnnn
```

## Examples

Example 1:

If you enter a LOCATE VMDBK command with the VMDBK keyword and specify a user ID pattern, you may see something like this:

```
locate vmdbk cptest*
Userid  VMDBK  Type      VMDBK-Real
CPTST1  05ADD000 ORIGIN    000000AA01899000
```

## LOCATE VMDBK

```
CPTTEST1 06BCF000 MP          00000208B6894000
CPTTEST2 07BDF000 ORIGIN     000000FC9CE33000
CPTTEST3 03BCE000 ORIGIN     0000009B7B527000
Ready;
```

This response shows that user ID CPTTEST1 is using two virtual processors.

Example 2:

If you enter LOCATE without the VMDBK keyword, you may see something like this:

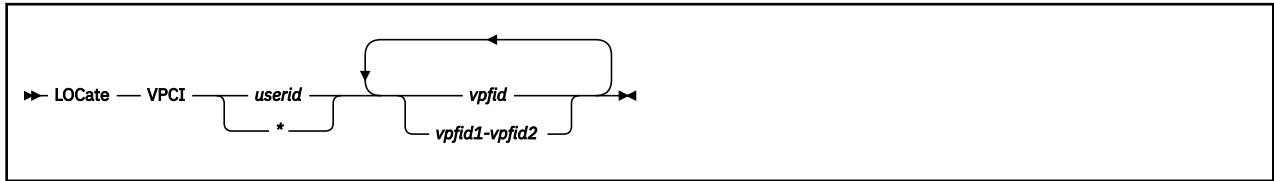
```
locate tools
VMDBK = 05ADD000 VMDBK-Real = 00000000_05ADD000
Ready;
```

Note that this response does not tell you if the TOOLS virtual machine is using multiple virtual processors. Note also that SET UNDERSCORE ON is in effect.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on

## LOCATE VPCI



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE VPCI to display the address of a user's virtual PCI function block and its associated control blocks.

### Operands

#### *userid*

\*

identifies the virtual machine to which the function or functions are currently attached. The specified user must be logged onto the system. If you are locating a VPCI for your own virtual machine, you may specify your user ID as an asterisk (\*).

#### *vpfid*

#### *vpfid1-vpfid2*

specifies a virtual PCI function ID or a range of PCI function IDs whose control block addresses are to be displayed. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. The total number of PCI functions specified cannot exceed 256.

### Responses

Response 1:

```
PCIF      VPCI      RPCI
xxxxxxxx  xxxxxxxx  xxxxxxxx
```

Where:

#### **PCIF**

shows the virtual PCI function ID

#### **VPCI**

shows the host logical storage address of the corresponding virtual PCI block.

#### **RPCI**

shows the real PCI function block address of the real PCI function associated with this virtual PCI function.

### Examples

Example 1:

If you enter a LOCATE VPCI \* 17 command, you may see something like this:

```
PCIF      VPCI      RPCI
00000017  012E8708  02E36288
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP022E A virtual PCI function ID was not supplied or it was invalid.
- HCP045E *userid* not logged on
- HCP5780E Adjunct does not support the *command subcommand* command
- HCP6000E The range of PCI function ids cannot exceed 256.
- HCP6720I No items found to satisfy selection criteria

## LOCATE VSMBK



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE VSMBK to display the address of a system VTAM service machine block (VSMBK).

### Operands

#### *userid*

specifies the user ID (or user ID pattern) whose VSMBK entries CP should locate. For more information about user ID patterns, see Usage Note “1” on page 647.

\*

tells CP to locate the VSMBK entry for your user ID.

### Usage Notes

1. Pattern matching follows the rules used by the CMS LISTFILE command. Use an asterisk (\*) to match any number of characters; use a percent symbol (%) to match any single character.

### Responses

Response 1:

```
Userid  VSMBK
userid  nnnnnnnn
:
```

#### **Userid**

shows the user ID you specified, or a user ID that matches the pattern you specified.

#### **VSMBK**

shows the host logical storage address of the VSMBK for that user ID.

### Examples

Example 1:

If you enter a LOCATE VSMBK command, you may see something like this:

```
locate vsmbk vtam
Userid  VSMBK
VTAM    0762B1C8
Ready;
```

### Messages

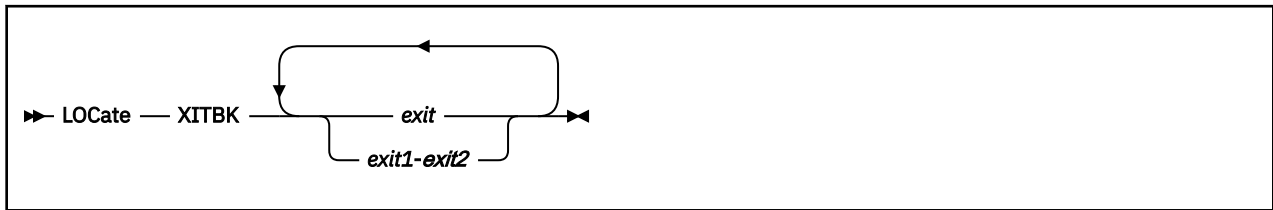
- HCP002E Invalid operand - *operand*

## LOCATE VSMBK

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP6720I No items found to satisfy selection criteria



## LOCATE XITBK



### Authorization

Privilege Class: C, E

### Purpose

Use LOCATE XITBK to display the address of the CP exit block for a specific exit point.

### Operands

*exit*

*exit1-exit2*

is the number of the CP exit block whose address you want to display. The exit number must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single exit number, a range of exit numbers, or any combination thereof.

### Usage Notes

1. CP creates CP exit blocks for exit points when you issue the ASSOCIATE EXIT command (see [“ASSOCIATE EXIT”](#) on page 20) to associate one or more entry point names or external symbols with that exit point. If no one issued the ASSOCIATE EXIT command, there is no CP exit block for CP to locate and CP issues error message HCP2752E.

After creating a CP exit block, CP will not erase that CP exit block until the next IPL. Disabling the exit point affects certain fields in the CP exit block, but does not erase it. Disassociating the entry point names and external symbols erases those fields in the CP exit block, but does not erase the CP exit block itself.

2. To display status and usage statistics information about a specific exit point, use the QUERY EXITS command (see [“QUERY EXITS”](#) on page 907).
3. To display the address of the CP indirect call locator block for a specific exit point, use the LOCATE ICLBK command (see [“LOCATE ICLBK”](#) on page 614).
4. For more information about user-defined exit points, see [z/VM: CP Exit Customization](#).

### Responses

Response 1:

```
Exit XITBK
nnnn nnnnnnnn
:
```

#### Exit

shows the exit number (exit point) you specified, or an exit number included in a range you specified.

#### XITBK

shows the host logical storage address of the CP exit block for that exit number.

### Examples

Example 1:

To display the address of the CP exit block for exit point X'070C', enter the following:

```
locate xitbk 70c
Exit XITBK
070C 005234C8
Ready;
```

Example 2:

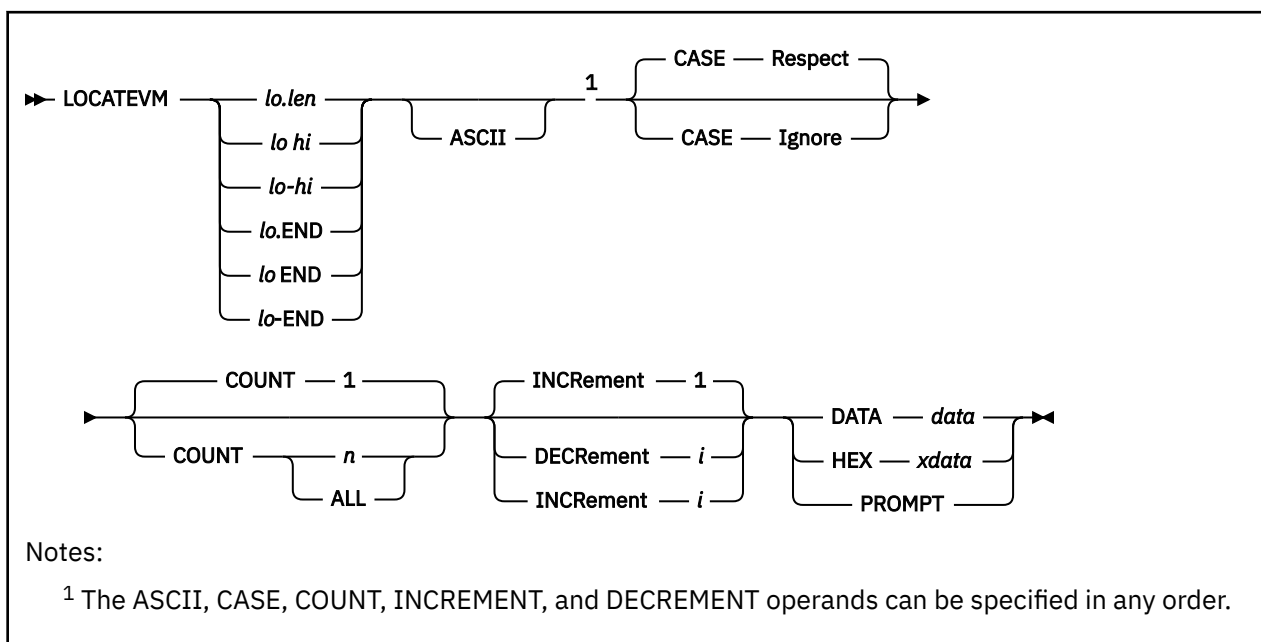
To display the addresses of the CP exit blocks for exit points X'0005' and X'F000', enter the following:

```
locate xitbk f000 5
Exit XITBK
F000 01F101C8
0005 061E19C8
Ready;
```

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP2752E Exit number *exit* does not exist
- HCP2752E Exit numbers *exit1-exit2* do not exist
- HCP6706E Invalid CP Exit number - *exit*

## LOCATEVM



### Authorization

Privilege Class: G

### Purpose

Use LOCATEVM to search for certain data in your virtual machine's storage.

### Operands

#### *lo.len*

specifies the address of the beginning of the search area and the number of bytes to its end. *lo* and *len* must both be hexadecimal numbers. You must enter a period between these two values.

#### *lo hi*

#### *lo-hi*

specifies the addresses for the beginning and end of the search area. *lo* and *hi* must both be hexadecimal numbers.

*lo*, *len*, and *hi* can be a 1-digit to 16-digit hexadecimal 64-bit address, separated with an optional underscore for readability.

#### *lo.END*

#### *lo END*

#### *lo-END*

specifies the address of the beginning of the search area and tells CP to search through to the end of the available storage. *lo* is a hexadecimal number. You may use any of the above formats.

#### CASE

tells CP to convert the data you enter to upper case before it is compared to the data in storage.

**Note:** Unless you are using DIAGNOSE code X'08' to send commands to CP in mixed case, CP converts commands entered from the terminal to upper case before it processes them. If you specify the HEX operand, CASE is ignored. The default is CASE RESPECT.

**Respect**

tells CP not to convert the data from the command and the data from storage before comparing them. RESPECT is the default.

**Ignore**

tells CP to convert the data from the command and the data from storage before comparing them so that the upper and lower case characters are considered equal.

**ASCII**

tells CP to convert the data you enter to ASCII before it is compared to the data in storage.

**COUNT *n*****COUNT ALL**

specifies how many matches to report. ALL reports all matches. The default is COUNT 1.

**DECRement *i***

tells CP to go *i* number of bytes backward through storage to search for the data. *i* must be a hexadecimal number. CP begins at the address you have specified as the end of the search area and compares the data you enter with what is at that address. If it does not find a match, it goes back *i* bytes and tries again. When it reaches the beginning of the search area or finds as many matches as you have specified with the COUNT keyword, CP displays the addresses of the fields within the search area that contain the data that you are looking for.

*i* can be a 1-digit to 16-digit hexadecimal 64-bit address, separated with an optional underscore for readability. If the underscore is used there must be 8 hexadecimal digits to the right of it.

**INCRement *i***

tells CP to go *i* number of bytes forward through storage to search for the data. *i* must be a hexadecimal number. CP begins at the address you have specified as the start of the search area and compares the data you enter with what is at that address. If it does not find a match, it advances *i* bytes and tries again. When it reaches the end of the search area or finds as many matches as you have specified with the COUNT keyword, CP displays the addresses of the fields within the search area that contain the data you are looking for.

*i* can be a 1-digit to 16-digit hexadecimal 64-bit address, separated with an optional underscore for readability. If the underscore is used there must be 8 hexadecimal digits to the right of it.

If you do not specify INCREMENT or DECREMENT, INCREMENT 1 is assumed.

**DATA *data***

specifies the data to search for. The data to be compared starts after the first blank following the keyword DATA. Leading, imbedded, and trailing blanks are included in the comparison. The length of the data to search for must be less than 254 bytes.

**HEX *xdata***

specifies the hexadecimal data to search for. You can specify the data as any number of strings of hexadecimal digits. If necessary, each string will be padded on the left with a single leading zero (0) to generate a string of an even number of hexadecimal digits. CP ignores blanks between the hexadecimal strings as it converts the strings for comparison. The length of the data to search for must be less than 254 hexadecimal digits.

If you specify the HEX operand, the CASE operand is ignored.

**PROMPT**

tells CP to prompt you to enter the data to be searched for. The length of the data to search for must be less than 254 bytes.

**Usage Notes**

1. CP only searches addresses for pages assigned. CP skips a page for an address not assigned (for example, an unreferenced page). If CP skips a page for any reason, the next comparison is in the next acceptable page at the address that would have been reached had the page or pages not been skipped and had all comparisons failed.

- If you specify the search area as *lo hi* or *lo-hi*, CP searches and compares the byte at address *hi*. For example,

```
locatevm 0 1 hex 00
```

causes CP to compare the two bytes (address 0 and address 1) to the specified data (00).

- Specifying the search area as *lo hi* or *lo-hi* is the same as specifying *lo.len* if  $len = hi - lo + 1$ .
- If CP finds a match, the address of the next comparison attempt is the address of the match plus the increment amount (or minus the decrement amount).
- If you specify DECREMENT, CP displays the addresses of matches with the highest address first.
- Whether you specify INCREMENT or DECREMENT, the set of addresses eligible for the search are the same. The difference is that INCREMENT causes the search to start from the *lo* address and progress upward, while DECREMENT causes the search to begin at the *hi* address and progress downward ending at the *lo* address. For example, in the following command,

```
locatevm 0 1000 decr c data 123456
```

the first comparison address is X'FF0'.

- If you specify PROMPT, you see the following:

```
Enter data now. To abort the command,
press ENTER with no data or press PA1.
```

- If a large search range is specified, the LOCATEVM command will take a large amount of CPU cycles and therefore can impact the overall system performance.

## Responses

Addresses in responses can be up to 16 hexadecimal digits, separated by an underscore if SET UNDERSCORE ON has been specified.

Response 1:

If you enter a LOCATEVM command, you might see something like this:

```
locatevm 1000-2000 hex 0
The data is located at 00001000.
Ready;
```

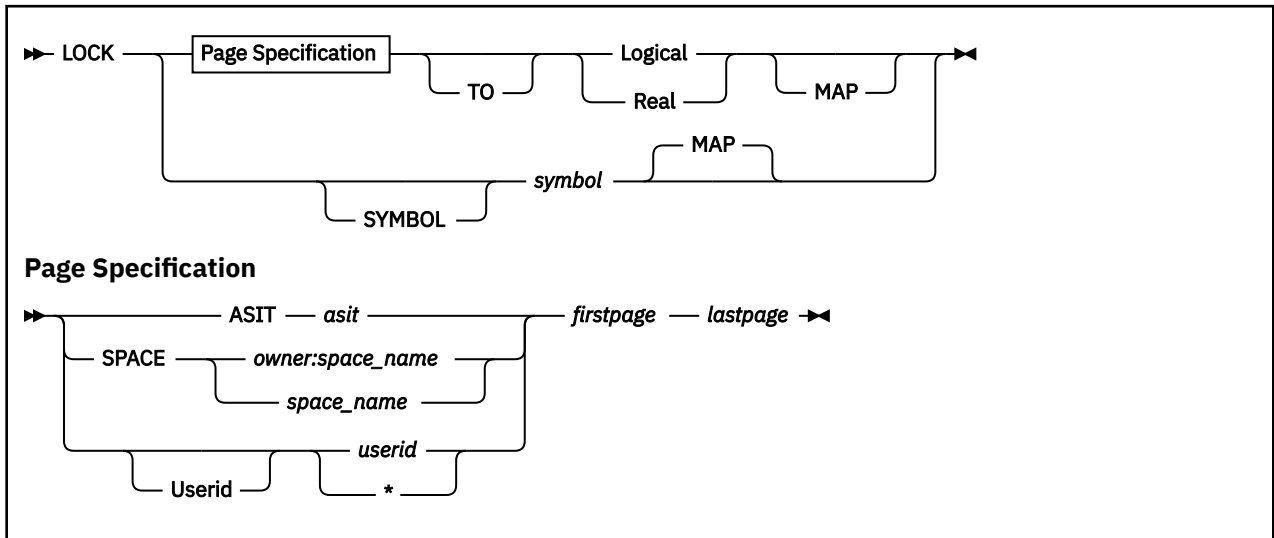
```
locatevm 4_00001000-4_00002000 hex 0
The data is located at 4_00001000.
Ready;
```

This response displays the address, in hexadecimal, of the information associated with your LOCATEVM request.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP004E Invalid hexloc *hexloc*
- HCP005E Invalid hexdata - *data*
- HCP009E Invalid range - *range*
- HCP009E An invalid address range was specified
- HCP6704E Missing token at end of line
- HCP6720I No items found to satisfy selection criteria
- HCP6896E The data field exceeds the maximum length of *nnn* characters.

# LOCK



## Authorization

Privilege Class: A

## Purpose

Use LOCK to keep selected guest pages in host storage.

## Operands

### ASIT *asit*

locks pages in the address space identified by the address-space-identification token, *asit*. The *asit* must be padded to 16 hexadecimal digits with leading zeros.

### SPACE *owner:space\_name*

locks pages in the address space identified by the space ID, *owner:space\_name*, where *owner* is the user ID of the virtual machine that owns the address space and *space\_name* is the name assigned to the address space.

If *owner:* is not specified, your virtual machine is assumed to be the owner of the address space.

The space name is a string of alphanumeric and national characters, 1 to 24 characters long. The space name "BASE" is preassigned by CP to be the name of the issuer's host-primary address space.

### Userid *userid*

#### *userid*

locks pages in the primary address space of the virtual machine identified by *userid*. The virtual machine must be logged on. If you are locking pages of your own virtual machine's storage, you can specify *userid* as an asterisk (\*).

### *firstpage*

is the hexadecimal number of the first user or address space page to be locked to host storage. Page numbers are assigned beginning with zero; each page is X'1000':

Storage Location	Page Number
0000-0FFF	0

<i>Table 29. Page Number Derivation (continued)</i>	
<b>Storage Location</b>	<b>Page Number</b>
1000-1FFF	1
2000-2FFF	2
⋮	⋮
1A000-1AFFF	1A
⋮	⋮

***lastpage***

is the hexadecimal number value of the last user or address space page to be locked to host storage. If only one page is to be locked, *firstpage* must be the same as *lastpage*.

**Logical**

locks the guest pages to host logical storage pages.

**Real**

locks the guest pages to real storage frames.

**MAP**

specifies that a map of the locked host storage should be displayed.

**SYMBOL *symbol***

has no effect and is retained only for compatibility purposes. All symbols are considered resident, which means they cannot be locked or unlocked.

**Usage Notes**

- In the primary address space, the first and last page numbers are treated as guest real addresses. They are prefixed with the prefix register value of the base virtual CPU of the target virtual machine. In other address spaces, the first and last page numbers are treated as *guest absolute* addresses. Prefixing is not applied to these addresses.
- If you are using LOCK for debugging purposes, use the LOGICAL operand to lock the guest pages to host logical storage. If you are using LOCK for performance reasons, use the REAL operand to lock the guest pages to real storage.
- If the SET RESERVED command has been issued by another virtual machine, you can use the LOCK command to enhance the efficiency of a particular virtual machine by keeping pages that are frequently referred to (such as page zero of the virtual machine's primary address space) in host storage. However, if the number of real storage frames available for paging is limited, you should not enter the LOCK command without the approval of your system programmer.
- If too many pages or frames of host storage are locked, there might not be enough resources available for the remaining virtual machines or for CP to operate efficiently. The performance of the system might be severely degraded because of the excessive contention for the available resources. Particular caution is warranted when locking pages in logical. Regardless of the real storage size, logical storage, also known as the system execution space, is limited to 2G and contains the CP nucleus, the prefix pages, dynamically allocated free storage, and other CP-use pages. Exhausting logical storage results in an SXA004 abend.
- Once a guest page is locked to host storage, it remains locked until one of these events occurs:
  - The CP UNLOCK command is issued for the page.
  - A SYSTEM CLEAR command is issued.
  - The page is released by a DIAGNOSE X'10' instruction or a TEST BLOCK instruction.
  - A MAPMDISK DEFINE operation is issued for the page.
  - A CP command causes an implicit system-reset-clear operation (for example, DEFINE STORAGE, LOGOFF, and some forms of the IPL command).

## LOCK

A locked page from an address space created by the ADRSPACE macro is unlocked when the address space is destroyed by the ADRSPACE macro.

6. Guest pages that are locked by this command are not unlocked by the operation of the [ADRSPACE ISOLATE](#) and [ADRSPACE PERMIT](#) macros. The ADRSPACE macros are described in *z/VM: CP Programming Services*.
7. The CP SYSTEM CLEAR command unlocks any locked pages and sets them to binary zeros.
8. The guest pages locked to host storage are 4 KB (4096-byte) blocks. This block of storage need not represent all of the user's virtual storage. The LOCK command may be issued as many times as required for one virtual machine to lock noncontiguous pages of storage. The remaining virtual machine storage blocks remain pageable.
9. If you omit the MAP operand from a LOCK command and you later want to find the address of the locked host logical storage page or real storage frame, reissue the same LOCK command with the MAP operand. No intervening UNLOCK command is required. Note that reissuing the same LOCK command does not put another lock on the guest page. Only one UNLOCK command is required to release the lock.
10. The LOCK and UNLOCK commands operate independently of the DIAGNOSE code X'98' page locking facility. Pages locked by the LOCK command cannot be unlocked by DIAGNOSE code X'98', nor can real I/O pages locked by DIAGNOSE code X'98' be unlocked by the UNLOCK command.
11. Some CP functions, such as DIAGNOSE code X'98' or X'214', may indirectly cause pages or frames of host storage to be unlocked.
12. If you specify the LOGICAL operand to lock a guest page to a host logical storage page, the corresponding real storage frame is also locked by CP. This makes it unnecessary to issue the LOCK command twice to lock the same guest page to both host logical storage and host real storage.  
  
Host storage must be unlocked in the same way it was locked. If you issued the LOCK command with the LOGICAL operand, you must also specify the LOGICAL operand on the UNLOCK command. That will release both the host logical storage page and the real storage frame.
13. The VMRELOCATE MOVE command is issued to relocate the guest to another system. Once a relocation unlocks pages, they remain unlocked even if the relocation is subsequently terminated or canceled prior to completion.
14. If you issue LOCK with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the LOCK command is rejected and error message HCPCPW033E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CLeAr option
  - SET MACHine (to a different machine architecture)

## Responses

**Note:** If SET UNDERSCORE ON has been specified, an underscore is used to separate 16-digit addresses for readability.

Response 1:

If you specify the LOGICAL and MAP operands, the response format is:

```
      VIRTPAGE  HOST LOGICAL  HOST REAL
userid  nnnnnnnn  nnnnnnnn  nnnnnnnnnnnnnnnn
:
Command complete
```



**userid**

is the user ID you specified. If you did not specify a user ID, this field is omitted.

**VIRTPAGE**

shows the number of the locked guest page.

**HOST LOGICAL**

shows the host logical storage address of the locked host page.

**HOST REAL**

shows the host real storage address of the locked frame.

Response 2:

If you specify the REAL and MAP operands, the response format is:

```

      VIRTPAGE  HOST REAL
userid  nnnnnnnn  nnnnnnnnnnnnnnnnn
:
Command complete

```

**userid**

is the user ID you specified. If you did not specify a user ID, this field is omitted.

**VIRTPAGE**

shows the number of the locked guest page.

**HOST REAL**

shows the host real storage address of the locked frame.

Response 3:

If you do not specify the MAP operand, the response is:

```

Command complete

```

**Examples**

Example 1:

For *firstpage* and *lastpage*, only the page numbers are specified. For example, to lock the first three pages of USERA's primary address space into host logical storage, enter:

```

lock userid usera 0 2 to logical
Command complete
Ready;

```

To get the MAP information while these pages are still locked, reissue the command with the MAP operand:

```

lock userid usera 0 2 to logical map

```

The response format will be similar to Example 4. This does not put an additional lock on these pages.

Example 2:

To lock the first three pages of USERA's address space named MYSPACE into host logical storage, enter:

```

lock space usera:myspace 0 2 to logical
Command complete
Ready;

```

Example 3:

## LOCK

To lock storage locations X'12000' through X'14FFF' of USERA's primary address space into real storage and display the addresses of the locked real storage frames, enter:

```
lock userid usera 12 14 to real map
      VIRTPAGE  HOST REAL
USERA  00012000  00000000078F8000
USERA  00013000  0000000006B7C000
USERA  00014000  0000000002A77000
Command complete
Ready;
```

Example 4:

To lock pages above 2 GB for USERA into host logical storage and display the addresses, enter:

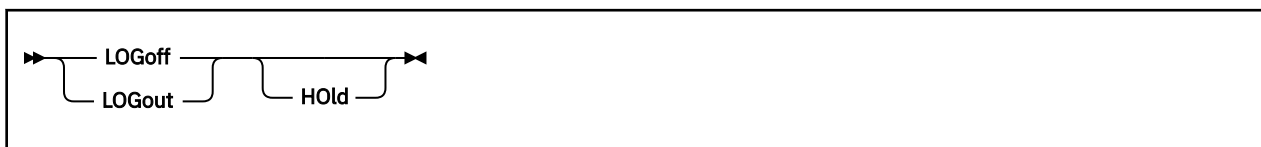
```
lock userid usera 123456 123457 to logical map
      VIRTPAGE          HOST LOGICAL  HOST REAL
USERA  00000001_23456000  078F8000      00000000_71A7F000
USERA  00000001_23457000  06B7C000      00000000_46B41000
Command complete
Ready;
```

Note that this example shows that SET UNDERSCORE ON is in effect.

## Messages

- HCP003E Invalid option - *option*
- HCP004E Invalid hexloc - *hexloc*
- HCP005E Invalid hexdata - *hexdata*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP033E Hexloc missing or invalid
- HCP045E *userid* not logged on
- HCP160E Hexloc *hexloc* exceeds storage
- HCP410E CP ENTERED; PAGING ERROR
- HCP1003E *module* was not found.
- HCP6720I No items found to satisfy selection criteria.

## LOGOFF



### Authorization

Privilege Class: Any

### Purpose

Use LOGOFF or LOGOUT to end a virtual machine session.

### Operands

#### HOLD

causes the non-SNA TTY terminal telecommunication connection to remain in effect after your session is disconnected. If you specify the HOLD option from a logical device, you log off from the logical device without losing the connection between the logical device and the system.

### Usage Notes

1. When the LOGOFF command is entered from the virtual machine console as the first or only command on a CP command line (that is, while TERMINAL MODE is CP, or CP READ is displayed or the command line begins with #CP), z/VM attempts to break out of certain wait conditions that might exist. Thus the LOGOFF command may be entered to cause breakout of certain virtual machine wait conditions.

The breakout attempt occurs when the LOGOFF command is entered, not when the command is processed. Command errors do not prevent the breakout attempt. In some cases, a breakout resets the virtual machine.

2. When you log off, all active spool files are closed, all virtual devices (including temporary minidisks and virtual disks in storage) are detached (virtual disks in storage are also destroyed if you are the last user), dedicated devices are detached, all Concurrent Copy (CONCOPY) sessions on your virtual machine are terminated, and an accounting record is created for your session.
3. The LOGOFF command performs a system-reset-clear function on your virtual machine as if the SYSTEM CLEAR command had been processed. The effects of this system-reset-clear function are described in detail in the description for the SYSTEM CLEAR command.
4. Always log off your display in addition to turning off the power; turning off your display is not the same as logging off.

If you turn off the power at your display without logging off, your virtual machine is logged off at some installation-defined grace period (usually 15 minutes) after the host system determines that your display has been turned off. If you again turn on the display, your virtual machine is disconnected. You can reconnect the display to your virtual machine by logging on within the grace period; if you do not log on within the interval, your virtual machine is automatically logged off.

5. If you specify the HOLD option on an SNA/CCS terminal, the command is not valid, and an error message is issued. If you specify the HOLD option on a non-SNA TTY display terminal, or a logical terminal, it is accepted and processed. The HOLD option is accepted but ignored on all other terminals.
6. If your installation has coded the E3270HLD feature in the system real I/O configuration file for your TTY display terminal connected to your DACU control unit 7171, the telecommunication connection always remains in effect, regardless of the HOLD option.

## LOGOFF

7. Users who are being logged off from the system will be given QUICKDSP designation to expedite the logoff of the user.
8. The LOGOFF command will fail if issued when VMRELOCATE is in progress.
9. When the LOGOFF process begins to release guest memory, the user becomes unknown to most of the Control Program and can be logged on again as a new instance while the release of memory continues. For a guest with a large amount of memory instantiated, the release process can take a considerable amount of time. Accordingly, unless there are sufficient additional resources available, it is advisable to avoid restarting a workload on the new instance that consumes substantial memory and instead to perform maintenance or other activities that occasioned the LOGOFF.

Several commands will identify a guest that is in the memory release phase of LOGOFF. For all these command responses, alphabetic characters in the guest user identifier will have been translated to lower case. QUERY NAMES will show the guest as LOF; LOCATE VMDBK with an operand that matches the lower case guest user ID will show the VMDBK type as LOGOFF; and INDICATE PAGING will show LOGOFF in a new STATUS column for the guest.

## Responses

Response 1:

```
CONNECT= hh:mm:ss VIRTCPU= mmm:ss.hs TOTCPU= mmm:ss.hs
LOGOFF AT hh:mm:ss zone weekday mm/dd/yy
```

### **CONNECT= hh:mm:ss**

is the actual clock time in hours, minutes, and seconds spent in the current session. The maximum value for this field is 99:59.59. A time value larger than the maximum will be indicated by this value.

### **VIRTCPU= mmm:ss.hs**

is the virtual processor time in minutes, seconds, and hundredths of seconds used in the current session. The maximum value for this field is 999:59.99. A time value larger than the maximum will be indicated by this value. If multithreading is enabled, this is reported as MT-1 equivalent time.

### **TOTCPU= mmm:ss.hs**

is the total processor time in minutes, seconds, and hundredths of seconds (including virtual and overhead) used in the current session. The maximum value for this field is 999:59.99. A time value larger than the maximum will be indicated by this value. If multithreading is enabled, this is reported as MT-1 equivalent time.

These times are either the elapsed time for the entire session or the elapsed time since the ACNT command was entered for this user.

### **LOGOFF AT hh:mm:ss**

is the current time in hours, minutes, and seconds of your logoff.

### **zone**

is the time zone.

### **weekday**

is the current day of the week.

### **mm/dd/yy**

is the date in month/day-of-month/year format.

The system also responds with one of the following, depending on the type of terminal being used:

<b><u>Type of Terminal</u></b>	<b><u>Message</u></b>
3270 family	Press ENTER or CLEAR key to continue
SNA/CCS terminal	Press ENTER key to continue

Pressing the appropriate key on your terminal causes the logo to appear on the terminal, which readies the terminal for the next session. For terminals not listed above, control automatically returns to CP, and the logo is displayed.

Response 2:

```
{type {rdev} } LOGOFF AS userid     USERS = nnnnn
  {      {ldev} }
  {luname}
  {SYSC }
```

is the response sent to the primary system operator when you log off. If your virtual machine was logged on using the AUTOLOG command, this response is sent to both the primary system operator and the user who issued AUTOLOG.

**type**

is the user's display type.

**rdev**

**ldev**

is the real or logical device number of the user's display.

**luname**

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

**SYSC**

identifies the system console.

**userid**

identifies the user who is logging off.

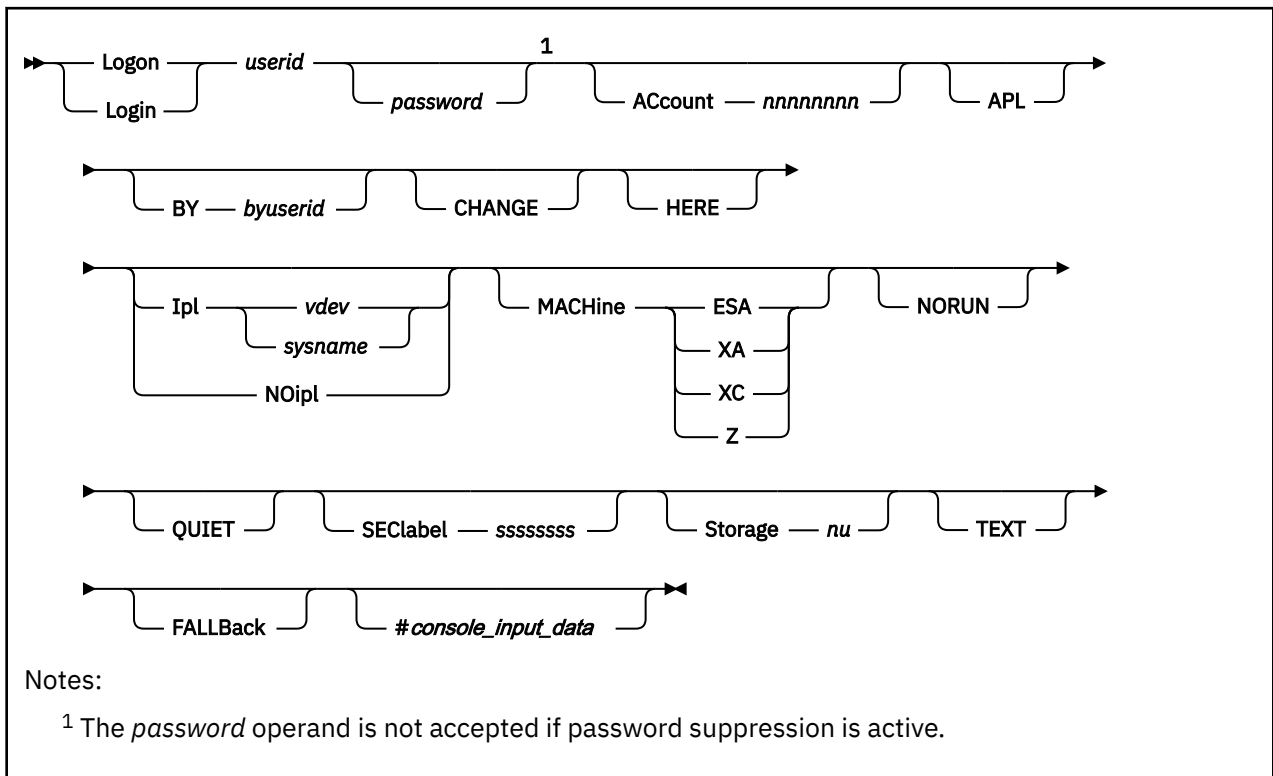
**USERS = *nnnnn***

is the number of users still on the system.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP6061I APPC wait canceled; virtual machine has been reset.

## LOGON



## Authorization

Privilege Class: Any

## Purpose

Use LOGON or LOGIN to access the virtual console of the specified virtual machine (*userid*).

## Operands

### *userid*

is the identifier assigned to you in the system directory.

### *password*

is the word or phrase that proves your identity to the system. If you omit *password*, the system will prompt you for it.

Passwords longer than 8 characters or that contain special characters such as blanks, called password phrases, are permitted only when an external security manager (ESM), such as RACF<sup>®</sup>, is used. Check with your system administrator to find out if an ESM is in use and if it is configured to allow password phrases. Refer to [“Usage Notes” on page 664](#) for more information about passwords.

### **ACcount** *nnnnnnnn*

specifies an alternate account number to be assigned to your virtual machine at logon time. This account number must be valid in order for the LOGON command to be completed successfully. If you do not specify this option, the primary (or first) account number specified in your entry in the system directory is used.

**APL**

specifies that upon successful completion of the user logon, the control program is to use an APL or TEXT translation table for translating console input and output rather than the standard translation table. This option of the LOGON command provides the same function as the TERMINAL APL ON or the TERMINAL TEXT ON commands.

**BY *byuserid***

specifies a BYUSER for the user ID. The password of the BYUSER is used for LOGON authorization checking for the user ID. The BYUSER must be listed on a LOGONBY statement in the directory entry for the user ID to use the BY operand. Refer to usage note [“11” on page 665](#) for information on ESM considerations.

**CHANGE**

specifies the user wishes to change their password. If your system is configured with an ESM, you may be able to use the CHANGE operand to enter into a change password dialog. If your system does not use an ESM or if the ESM does not support the CHANGE operand, the operand is ignored.

**HERE**

specifies that if this user ID is already logged on, it should be disconnected from its current terminal and reconnected at the terminal where this logon is requested.

**Ipl *vdev*****Ipl *sysname***

specifies the virtual device number or named saved system that you want IPLed upon completion of your logon processing. This option overrides the IPL statement (if any) specified in your entry in the system directory. The validity of the IPL operand you specify is determined by the IPL command processor after your logon is completed; it is not verified as part of the LOGON command processing.

**NOipl**

specifies that the IPL device specified in your entry in the system directory should not be used for an automatic IPL.

**MACHine ESA****MACHine XA****MACHine XC****MACHine Z**

specifies the mode (architecture) of the virtual machine. This option overrides the MACHINE statement (if any) that is specified in your entry in the system directory. For further information on the MACHINE directory statement, see [z/VM: CP Planning and Administration](#).

**ESA** designates ESA/390 (31-bit) architecture. An ESA virtual machine initially operates in ESA/390 architecture; the guest can switch the virtual machine between ESA/390 and z/Architecture.

**XA** is supported for compatibility and is functionally equivalent to ESA.

**XC** designates ESA/XC architecture. An XC virtual machine initially operates in ESA/XC architecture; the guest can switch the virtual machine between ESA/XC and z/XC.

**Z** designates a guest that executes only in z/Architecture mode.

See usage note [“22” on page 666](#).

**NORUN**

specifies that upon successful completion of the user logon, the virtual machine is placed in a terminal CP READ state, and RUN is set OFF.

**QUIET**

suppresses the display of all informational LINK messages, system log messages, and information regarding spooled virtual reader, printer, and punch files.

**SEClabel *sssss***

specifies that the user is requesting to logon with a specific security label, identified by *sssss*. *sssss* is specified as a 1- to 8-character alphanumeric value. This operand is valid only when an ESM is installed and security label checking is enabled. For additional information, contact your security administrator.

**Storage nu**

specifies an alternate storage size to be assigned to your virtual machine. The storage size is specified in the form *nu*, where *n* is a 1- to 8-digit decimal number and *u* is the 1-character storage unit suffix:

*Table 30. Maximum Input Values for Storage Units*

Storage unit	Suffix ( <i>u</i> )	Maximum input value ( <i>n</i> )
Kilobytes	K	99999999
Megabytes	M	99999999
Gigabytes	G	99999999
Terabytes	T	16777216
Petabytes	P	16384
Exabytes	E	16

**Notes:**

1. A K specification is rounded up to a MB value.
2. The maximum input value of 99999999 for the K, M, or G suffix is not a size limit but the physical limit of the operand (8 digits plus suffix). If the maximum input value for one of these suffixes does not allow you to define the amount of storage you want, you need to use a larger storage unit.
3. An ESA/XC virtual machine can address up to 2 GB of storage in its base address space.
4. The value you specify for this option must be valid in order for the logon to be completed successfully. If you do not specify a value, the storage size defined in the user directory entry for your virtual machine is used. The maximum allowed value is the maximum storage size defined in the user directory entry for your virtual machine. The maximum cannot exceed a processor-specific limit imposed at run time.

**TEXT**

specifies that upon successful completion of the user logon, the control program is to use an APL or TEXT translation table for translating console input and output rather than the standard translation table. This option of the LOGON command provides the same function as the TERMINAL APL ON or the TERMINAL TEXT ON commands.

**FALLBack**

specifies that the user wants to perform a fallback logon.

This parameter is only relevant if an external security manager (ESM) is installed and the ESM supports this operand. Further processing may depend on ESM configuration, user authorization, or other factors. Refer to your ESM documentation for further information.

**#console\_input\_data**

is data that you want used as virtual console input in response to the first read your virtual machine issues. You may use the remainder of the command line to enter console data. The entire line, including any logical-line-end characters (#), is passed to your virtual machine.

**Note:** # refers to whatever logical-line-end character is the current system default.

**Usage Notes**

1. Upon a successful logon, CP creates your virtual machine configuration based upon information contained in the system directory.
2. Except for recognizing the system default logical-line-end character as the delimiter preceding a console\_input\_data operand, line-editing is not performed on any of the fields of the system logon screen or on responses to prompts before logon completes successfully.
3. A virtual machine is allowed to log on even though the number of users already logged on is equal to or greater than the maximum allowed by the MAXUSERS operand of the FEATURES system



configuration statement or by the SET MAXUSERS command if the user directory for the virtual machine was created using the IGNMAXU operand of the OPTION directory statement.

4. For compatibility, z/VM continues to accept the XA virtual machine designation. However, whether the virtual machine is defined as XA or ESA makes no difference when running on z/VM. A virtual machine defined as XA has the capabilities of an ESA virtual machine and is considered to be an ESA virtual machine simulating the ESA/390 architecture (or z/Architecture, if a guest has issued the instruction to change the architecture of the virtual machine).
5. CP does not support System/370 architecture (370 mode) virtual machines. If a LOGON is attempted for a virtual machine whose directory entry specifies 370 mode, the LOGON command is completed, but the virtual machine is set to XA mode instead of 370 mode. Message HCP1512E is issued to inform the user.
6. If a LOGON is attempted for a virtual machine whose directory entry specifies z/Architecture-only mode, but the relocation domain does not support that mode, the LOGON command is completed, the virtual machine is set to ESA mode instead, and message HCP1512E is issued to inform the user.
7. If you were disconnected from your virtual machine because of a console error, you have an installation-defined grace period (usually up to 15 minutes) to log on again. If you do not log on within the grace period, your virtual machine is automatically logged off. If this happens, you may have to reconstruct files and restart jobs that were interrupted by the error.
8. When you enter the LOGON command to reconnect to your virtual machine, the options NOIPL, IPL, ACCOUNT, MACHINE, SECLABEL, and STORAGE are ignored. If you enter any of these options, you receive an informational message and the value is not changed. The console-input data line is also ignored.
9. If your virtual machine is running in disconnected mode, and you enter the LOGON command to reconnect your console, certain SET and TERMINAL command operands are forced OFF. You can enter the QUERY SET and QUERY TERMINAL commands to check which functions are currently active.
10. If you or a system administrator has used the COMMAND directory statement to specify commands that will be automatically issued when a virtual machine is logged on, be aware that you may receive additional responses as a result of those commands whenever you log on. See the [COMMAND directory statement](#) in *z/VM: CP Planning and Administration* for more information.
11. When an ESM such as RACF is installed, it may have other authorization criteria defined that completely replace those provided by the LOGONBY directory statement. Refer to the documentation provided with your ESM for more information.
12. Users defined with the LBYONLY operand in the password field of their USER or IDENTITY statement in the system directory may be restricted from performing functions that require password validation. If an ESM is not installed, a LOGON of a user defined with the LBYONLY operand can be done only by an authorized user using the BY option of the LOGON command. A user defined with the LBYONLY operand is not allowed to log on to any user ID using the LOGON command with the BY option. If an ESM is installed, refer to the documentation provided with your ESM for more information.  
  
Refer to the [USER Directory Statement](#) or [IDENTITY Directory Statement](#) in *z/VM: CP Planning and Administration* for more details on the LBYONLY operand.
13. If your installation is using the password suppression facility for the LOGON command (controlled by the PASSWORDS\_ON\_CMDs operand of the FEATURES system configuration statement or by the SET PASSWORD command), you cannot specify your password on the command line. You must wait for the ENTER PASSWORD prompt before you type in the password. The password is automatically masked, thus improving system security. If your installation is not using the password suppression facility for the LOGON command, you can specify your password on the command line, or you can wait for the ENTER PASSWORD prompt before you type in the password.
14. If the password suppression facility is not active for the LOGON command, allowing passwords to be specified on the command, user logon passwords defined in the system directory cannot be identical to any of the LOGON command options.
15. User IDs that are defined with the NOPASS operand are able to logon without password authorization, unless an ESM is installed. When an ESM is installed, password authorization may

be required despite the NOPASS operand. Refer to the documentation provided with your ESM for more information.

16. Passwords are restricted to a maximum of 8 characters in length unless an ESM is used. In that case, the ESM may permit passwords up to 200 characters in length. The RACF Security Server for z/VM supports passwords up to 100 characters in length. Refer to the documentation provided with your ESM to determine the maximum.
17. You must surround the password with single quotation marks or omit it entirely when it begins with a single quotation mark, begins or ends with a blank, or is entered on the command line and contains one or more blanks. You must use double quotation marks that are part of the password when the password is quoted. See [“Examples” on page 669](#).
18. Long passwords and password phrases can contain any character between X'00' and X'FF'. This range includes all upper and lower case characters, blanks, and special characters. Using characters less than X'40' is not recommended. Even though CP supports all characters, your ESM may restrict the use of some characters.
19. The disconnect performed when the HERE option is specified may take significant time to complete because of the attempt made to process the disconnect even in the case of a hung terminal.
20. The NORUN option is intended when RUN is set ON for users who wish to reconnect and get into CP READ. However, the option is always accepted with no effect in cases other than when RUN is set ON. The option is particularly helpful for users who need to enter CP commands and who are in full-screen mode with no break-key and with RUN set ON. These users can power-off their terminals to be disconnected and then reconnect using LOGON with NORUN.
21. When a z/VM system is running in a Single System Image (SSI) configuration, and an External Security Manager (ESM) has enabled checking of Mandatory Access Controls (MAC), the SECLABEL operand is not allowed for virtual machines defined by an IDENTITY statement in the z/VM User Directory.
22. An ESA (or XA) virtual machine will be put into either full ESA/390 mode or ESA/390 compatibility mode, depending on the level of ESA/390 capability available in the machine where the virtual machine is logged on or in the virtual machine's relocation domain. ESA/390 compatibility mode allows a subset of ESA/390 functionality sufficient for CMS and GCS, but lacks advanced functions like dynamic address translation (DAT). For details on ESA/390 compatibility mode, see [z/Architecture Principles of Operation](#).

## Responses

Response 1:

- 3270 Display Version:

```
ENTER PASSWORD (It will not appear when typed):
```

This version of the password prompt is used for 3270 family displays. It indicates that your user ID has been accepted. Enter your password.

- Line-Mode ASCII Version:

```
ENTER PASSWORD:
```

This version of the password prompt is used for line-mode ASCII terminals. It also indicates that your user ID has been accepted. Enter your password.

Response 2:

```
z/VM Version v Release r.m, Service Level yynn (64-bit),  
built on IBM Virtualization Technology
```

**v**

is the software version level.

**r.m**

is the release and modification level.

**yynn**

is the service level which identifies the most recent RSU service tape that has been applied, where *yy* is the last two digits of the year and *nn* is the sequential number of the RSU tape for that year.

Response 3:

```
LOGMSG - hh:mm:ss zone day-of-week mm/dd/yy
```

indicates the time and date at which the system log message was recently revised and is displayed only when there are log messages to be displayed. All lines of the log message for which the first character is an asterisk are displayed. If you wish to see all of the system log messages, enter the QUERY LOGMSG command.

**hh:mm:ss**

is the time when the log message was last updated.

**zone**

is the time zone.

**day-of-week**

is the day when the log message was last updated.

**mm/dd/yy**

is the month/day/year when the log message was last updated.

Response 4:

```
FILES: nnnn RDR, nnnn PRT, nnnn PUN
       NO   NO   NO
```

where *nnnn* is the number of spool files of each type.

Response 5:

One of the following appears, indicating the time, day of the week, and date at which the logon or reconnect is complete.

```
LOGON AT hh:mm:ss zone weekday mm/dd/yy
```

```
RECONNECTED AT hh:mm:ss zone weekday mm/dd/yy
```

For the value of the variable fields, see Response 2.

Response 6:

One of the following is the response to the primary system operator.

```
{type {rdev}} LOGON AS userid USERS = nnnn [FROM ipadr]
{
  {ldev}}
  {lname}
  {SYSC}
```

```
{type {rdev}} RECONNECT userid USERS = nnnn [FROM ipadr]
{
  {ldev}}
  {lname}
  {SYSC}
```

```
{type {rdev}} LOGON AS userid USERS = nnnn BY byuserid [FROM ipadr]
{
  {ldev}}
  {lname}
  {SYSC}
```

```
{type {rdev}} RECONNECT userid USERS = nnnn BY byuserid
{
  {ldev}}
  {lname}
  {SYSC}
```

## LOGON

**type**

is the display type.

**rdev****ldev**

is the real or logical device number of the user's display.

**luname**

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

**SYSC**

identifies the system console.

**userid**

identifies the user ID of the virtual machine that just completed logging on to the system.

**USERS = nnnnn**

is the number of users on the system.

**FROM ipadr**

is the IP address.

An IPv4 address is displayed in dotted-decimal format.

**Example:** 9 . 130 . 44 . 144

An IPv6 address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros).

**Example:** The following is an IPv6 address:

```
F802:0000:0000:0000:0000:0000:0A00:0001
```

This address is displayed in compressed format as:

```
F802::A00:1
```

**BY byuserid**

identifies the user ID that logged on the virtual machine. This information is only provided when the LOGON command is entered with the BY operand.

Response 7:

```
DISCONNECT AT hh:mm:ss zone weekday mm/dd/yy BY LOGON FROM {type}{rdev}}
                                                              {
                                                              {ldev}}
                                                              {luname}
                                                              {SYSC}
                                                              }
```

is the response sent to the terminal from where your user ID is disconnected prior to processing a reconnect.

**type**

is the display type.

**rdev****ldev**

is the real or logical device number of the terminal from where the LOGON was entered with the HERE option.

**luname**

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

**SYSC**

identifies the system console.

Response 8:

```
DISCONNECTED FROM {type} {rdev}}
                   {
                   {ldev}}}
```

```
{luname }
{SYSC }
```

is the response you receive when your user ID is disconnected from another terminal prior to reconnecting.

**type**

is the display type.

**rdev****ldev**

is the real or logical device number of the terminal from where the user was disconnected.

**luname**

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

**SYSC**

identifies the system console.

Response 9:

```
{type{rdev}} DISCONNECT userid USERS=nnnnn BY LOGON FROM {type{rdev}}
{ {ldev}} { {ldev}}
{luname } {luname }
{SYSC } {SYSC }
```

is the response sent to the primary system operator when a disconnect for your user ID is performed prior to a reconnect.

**type**

is the display type from where the user was disconnected.

**rdev****ldev**

is the real or logical device number of the terminal from where the user was disconnected.

**luname**

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

**userid**

is the user ID of the virtual machine for which the disconnect and log on was requested.

**USERS = nnnnn**

is the number of users remaining on the system after the disconnect is performed

**FROM type**

is the display type from where the LOGON command was issued.

**rdev****ldev**

is the real or logical device number of the terminal from where the LOGON command was entered.

**luname**

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

**SYSC**

identifies the system console.

**Examples**

Example 1:

If you enter LOGON with a password that contains one or more blanks, you must surround it in single quotation marks to preserve any mixed-case characters and imbedded blanks.

```
logon alan 'now is the time for all good men' noip1
```

Example 2:

When you attempt to enter a password containing one or more single quotation marks, they must be entered as 2 single quotation marks.

```
logon mike 'That' 's Great!'
```

## Messages

- HCP003E Invalid option - *option*
- HCP007E Invalid userid - *userid*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP024E Cylinders or blocks missing or invalid
- HCP025E Storage missing or invalid
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP045E *userid* not logged on
- HCP046E *type rdev* offline
- HCP050E LOGON unsuccessful--incorrect userid and/or password
- HCP052E Error in CP directory
- HCP054E Already logged on {GRAF *rdev*|GRAF *ldev*|*luname*|DISC|SYSTEM *sysname*|SYSC}
- HCP054E LOGON not permitted because the SSI cluster is not in STABLE mode.
- HCP054E LOGON not permitted because the system has not completed initialization.
- HCP067E Command format not valid
- HCP067E Command format not valid. SECLABEL operand is not allowed for this user.
- HCP090I Device *vdev* not defined; {*rdev/volid*} not available
- HCP091E DASD *vdev* not defined; temp space not available
- HCP093E Storage size requested *reqsize* exceeds maximum allowed on this processor *maxsize*. Size set to maximum allowed.
- HCP094E Storage exceeds allowed maximum of *maximum* {M|K}
- HCP095I Device *vdev* is not defined because the maximum number of devices is already defined.
- HCP101E DASD *vdev* forced R/O; R/O by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[:; stable by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP102E DASD *vdev* forced R/O; R/W by {*nnnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[:; stable by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP103E DASD *vdev* forced R/O; R/W by {*nnnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}, R/O by {*nnnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[:; stable by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP104E *userid vdev* not linked; R/O by {*nnnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[:; stable by {*nnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]|exclusive by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]
- HCP105E *userid vdev* not linked; R/W by {*nnnnn* users|*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}[:; stable by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]|exclusive by {*userid*|*userid* at *sysname*|*userid* ADJUNCT|*userid* ADJUNCT at *sysname*}]

- HCP106E *userid vdev* not linked; R/W by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}, R/O by {*nnnnn users|userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname* } [; stable by {*userid|userid at sysname|userid ADJUNCT|userid ADJUNCT at sysname*}]
- HCP107E *userid vdev* not linked; not in CP directory
- HCP108E *userid vdev* not linked; valid *valid* not mounted
- HCP109E *userid vdev* not linked; {Excessive link indirections|invalid link device}
- HCP110E *userid vdev* not linked; *type vdev* already defined
- HCP117E Userid *userid* not linked; valid *valid* conflict
- HCP174E Paging I/O error; IPL failed
- HCP196I Secondary user not in CP directory
- HCP205I The virtual=real area is not available because it was not generated.
- HCP264I One or more options are ignored during reconnect processing – *option(s)*
- HCP266E An account number was not supplied or is invalid.
- HCP288E LOGON from the initial screen was unsuccessful
- HCP298E *userid vdev* not linked; request denied
- HCP361E LOGOFF/FORCE pending for user *userid*
- HCP365I Requested language *langid* is unavailable. Language *langid* set. RC=*rc*
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP475I Fatal I/O error trying to read directory from *valid* { for user *userid*}
- HCP799E IPL failed due to insufficient or faulty storage
- HCP844E The device is not defined due to an invalid control unit type.
- HCP849E A *userid* must be specified with the spoolid
- HCP1015E Insufficient storage is available to satisfy your request.
- HCP1128E FCP *vdev* not attached; USERACCESSID not supported on this device.
- HCP1150E DASD {*rdev|vdev*} is not a valid base exposure.
- HCP1151E *userid vdev* has not been linked because it is not within CP volume cylinder extents.
- HCP1152E *userid vdev* has not been linked because it would overlap system {spool|paging|temporary disk} space
- HCP1401I Expanded storage is not supported.
- HCP1454E CPU *nn* does not exist.
- HCP1501E An accounting error was encountered.
- HCP1502E The directory entry for this *userid* is in use.
- HCP1505I The device being used for LOGON conflicts with the console definition in the directory.
- HCP1510I Device *vdev* not defined; *device vdev* already defined
- HCP1512E The System/370 machine mode was requested, but CP does not support System/370 mode virtual machines; the machine mode will be set to XA instead.
- HCP1512E The z/Architecture-only machine mode was requested, but is not supported in this relocation domain. The machine mode will be set to ESA instead.
- HCP1513E Log on already in progress on {GRAF *rdev*|GRAF *ldev*|*luname*|SYSC}
- HCP1558E Command processing cannot complete.
- HCP1615E IPL LOADDEV|DUMPDEV failed because not all necessary parameters have been set.
- HCP1700I One or more virtual CPUs are not defined due to an error in the CP directory.
- HCP1830I Virtual architecture has been adjusted to permit AP Dedicated Crypto facilities not allowed in the assigned or default relocation domain.

## LOGON

- HCP2094I Requested ABSOLUTE I/O priority range falls outside range currently allowed by CP and has been changed to fit within CP's range.
- HCP2550I The console has been defined as a 3215 because a console mode of 3270 is not supported by the VTAM Service Machine controlling the real terminal.
- HCP2551E The console has been defined as a 3215 because the real device is not a 3270 display.
- HCP2553E TERMINAL BRKKEY *name* is not allowed during this session; it has been changed to PA1.
- HCP2554E TERMINAL BREAKIN GUESTCTL is not allowed during this session; it has been changed to IMMED.
- HCP2808E User ID is a CF Service Machine and may not be started by the LOGON command.
- HCP2880E *userid vaddr* is not linked, CSE LINK I/O error, code *cc*.
- HCP3224I NICDEF network configuration is ignored due to the current setting of VMLAN DNA.
- HCP6050E Your *userid* is not authorized to logon *userid userid*.
- HCP6053E You cannot issue the {LOGON|AUTOLOG|XAUTOLOG} command because the maximum number of users allowed is already logged on.
- HCP6059I One or more devices were not defined because an error occurred while reading the CP directory.
- HCP6525E External Security Manager is unavailable.
- HCP6704E Missing token at end of line



## MESSAGE

---

### Purpose

Use the MESSAGE commands to transmit message text to the virtual consoles of other active users.

### Usage Notes

(General for MESSAGE)

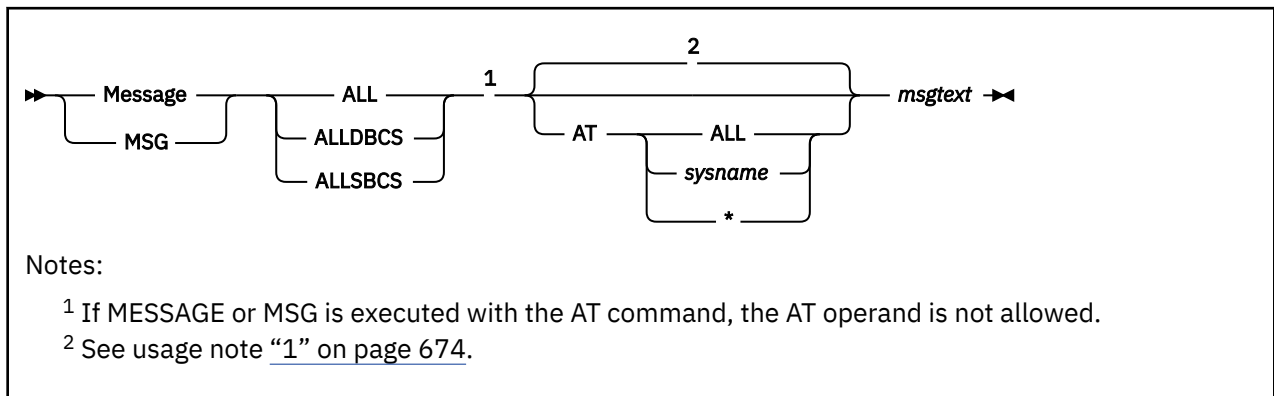
1. If a message is issued from a CMS environment, the command and the message text cannot be longer than 240 characters.  
If a message is issued from a CP environment, the command and message length is limited by the input area of the terminal.
2. If you send a message to a user who is not logged on and has not defined a secondary user, or to a user who has suppressed the receiving of messages (that is, a user who has issued the SET MSG OFF command), your message is not transmitted, and you receive an informational message. The message that was transmitted is not saved. You must send it again when the user is receiving messages.
3. The MESSAGE command routes the message to the secondary user of a disconnected virtual machine. If the secondary user is connected or the secondary user is disconnected but has a valid path to a \*MSG or \*MSGALL CP system service, the secondary user receives the message.  
If the secondary user is disconnected and there is no valid path to a \*MSG or \*MSGALL CP system service, *userid* not receiving; disconnected. is returned to the command issuer.
4. Messages are displayed only when the display device is ready to receive output. If you have a CP READ or a VM READ, messages are held until you end the read. Note, however, that if you are using an SNA/CCS terminal, messages are displayed when in CP READ or VM READ.
5. If you receive a message on a 3270 virtual console that has the alarm feature, the alarm sounds.
6. DBCS text is supported as message text for the MESSAGE commands.

### Messages

(General for MESSAGE)

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP057I *userid* not receiving; *text*
- HCP1142E The AT option is not allowed if this command is initiated with an AT command from another system.
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.
- HCP6600E An error was detected by installation-wide {*CP module name*|exit point 1210} - return code *nnnn*

## MESSAGE ALL



### Authorization

Privilege Class: A, B

### Purpose

Use MESSAGE ALL or MSG ALL to transmit message text to all active users and disconnected users.

### Operands

#### ALL

transmits the message to all users.

#### ALLDBCS

transmits the message to all users logged on to DBCS (double-byte character set) terminals.

#### ALLSBCS

transmits the message to all users and all disconnected users logged on to SBCS (single-byte character set) terminals.

#### AT ALL

transmits the message to all systems in the SSI cluster to which this system belongs. The lowest return code is returned to the command issuer.

#### AT *sysname*

transmits the message to a specific system in the SSI cluster to which this system belongs. The return code is returned to the command issuer.

#### AT \*

transmits the message only to the system where the command is issued. This is the default if the system is not a member of an SSI cluster.

#### *msgtext*

is the text of the message you are sending. The length of the message is limited by the number of characters remaining on the input line after the command and appropriate operand are entered. You must include at least one blank between the ALL operand and the message text. DBCS text is supported as message text.

### Usage Notes

- If the AT operand is not specified:
  - If MESSAGE or MSG is not executed with the AT command:
    - If the sender's system is not a member of an SSI cluster, the default is AT \*.

- If the sender's system is a member of an SSI cluster, the default is AT ALL, except in the following cases:
    - If the sender is defined by a multiconfiguration virtual machine definition, the default is AT \*.
    - If a potential recipient user ID is defined by a multiconfiguration virtual machine definition, the default for that user ID is AT \*.
  - If MESSAGE or MSG is executed with the AT command, the AT operand function is bypassed (there is no default), and the message is sent to the system specified on the AT command.
2. The total number of logged-on users significantly affects command processing. The command might take a long time to process and might tie up user terminals.
  3. For additional information, see [“MESSAGE” on page 673](#).

## Responses

```
hh:mm:ss * MSG FROM userid : msgtext
```

### ***hh:mm:ss***

identifies the time the message was sent to the user, in hours, minutes, and seconds.

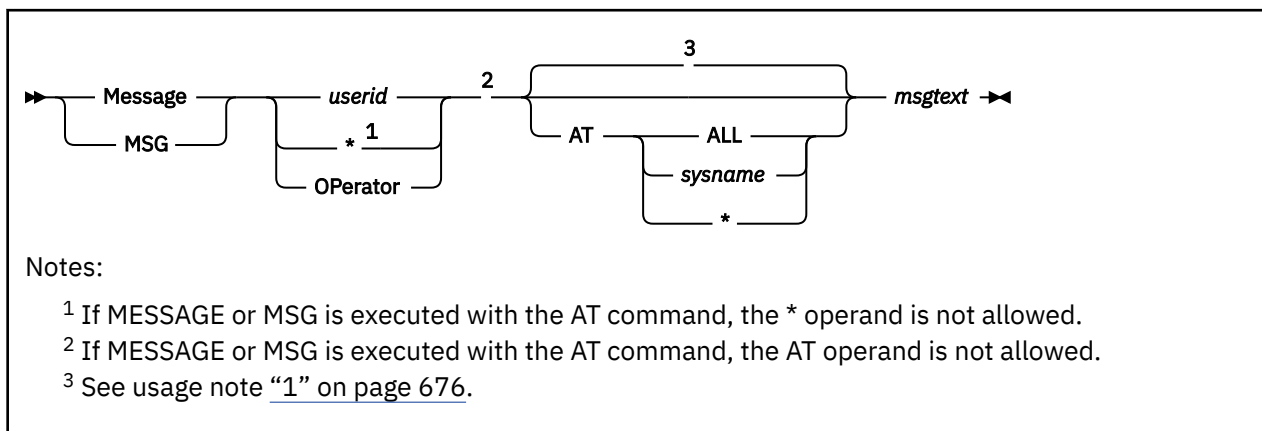
### ***userid***

identifies the user sending the message.

### ***msgtext***

identifies the message text.

## MESSAGE (User)



### Authorization

Privilege Class: Any

### Purpose

Use MESSAGE (User) or MSG (User) to send message text to a specified user ID or to the primary system operator.

### Operands

*userid*

\*

**OPerator**

identifies the user to whom you are sending a message. To send a message to yourself, specify an asterisk (\*).

Specify OPERATOR if you want to send a message to the primary system operator, regardless of the operator's user ID.

**AT ALL**

transmits the message to all systems in the SSI cluster to which this system belongs. The lowest return code is returned to the command issuer.

**AT *sysname***

transmits the message to a specific system in the SSI cluster to which this system belongs. The return code is returned to the command issuer.

**AT \***

transmits the message only to the system where the command is issued. This is the default if the system is not a member of an SSI cluster.

*msgtext*

is the text of the message you are sending. The length of the message is limited by the number of characters remaining on the input line after the command and appropriate operand are entered. DBCS text is supported as message text.

### Usage Notes

1. If the AT operand is not specified:
  - If MESSAGE or MSG is not executed with the AT command:

- If the sender's system is not a member of an SSI cluster, the default is AT \*.
  - If the sender's system is a member of an SSI cluster, the default is AT ALL, except in the following cases, when the default is AT \*:
    - If the specified user ID is OPERATOR.
    - If the specified user ID is logged on to the same system as the sender.
    - If the specified user ID or the sender is defined by a multiconfiguration virtual machine definition.
  - If MESSAGE or MSG is executed with the AT command, the AT operand function is bypassed (there is no default), and the message is sent to the system specified on the AT command.
2. You do not have to be logged on to issue the MESSAGE (User) command. However, if an external security manager (ESM) is installed on your system, the ability to send pre-logon messages might be disabled.
  3. If an ESM is installed on your system, you might not be authorized to use this command even after you have logged on. However, messages sent to or from the system operator and messages sent with the ALL option are not subject to authorization checking by the ESM. For additional information, contact your security administrator.
  4. The expression msg \* followed by text can be used to identify the current user ID of an abandoned display.
  5. For additional information, see [“MESSAGE” on page 673](#).

## Responses

Response 1:

```
hh:mm:ss * MSG FROM OPERATOR: msgtext
```

is the response received by the user when the system operator sends a message. For the contents of the variable fields, see under Response 2.

Response 2:

```
hh:mm:ss * MSG FROM
{userid }: msgtext
      {LOGNxxxx}
      {LOGLxxxx}
      {LOGVxxxx}
      {LOGMxxxx}
```

is the response received by the user when another user sends a message.

**hh:mm:ss**

identifies the time the message was sent to the user in hours, minutes, and seconds.

**userid**

**LOGNxxxx**

**LOGLxxxx**

**LOGVxxxx**

**LOGMxxx**

identifies the user sending the message. LOGNxxxx identifies the user in the process of logging on to a real device. LOGLxxxx identifies the user in the process of logging on to a logical device. LOGVxxxx identifies the logical unit name of the SNA/CCS terminal to which a user is in the process of logging on; it is the name from the VTAM definition of the terminal as an SNA logical unit. If the user is not logged on, LOGNxxxx or LOGLxxxx identifies the line number of the device from which the message originated. LOGMxxx identifies the user in the process of being relocated on to this system from another member of the SSI cluster. Userids starting with LOGN, LOGL, and LOGV are restricted.

**xxx or xxxx**

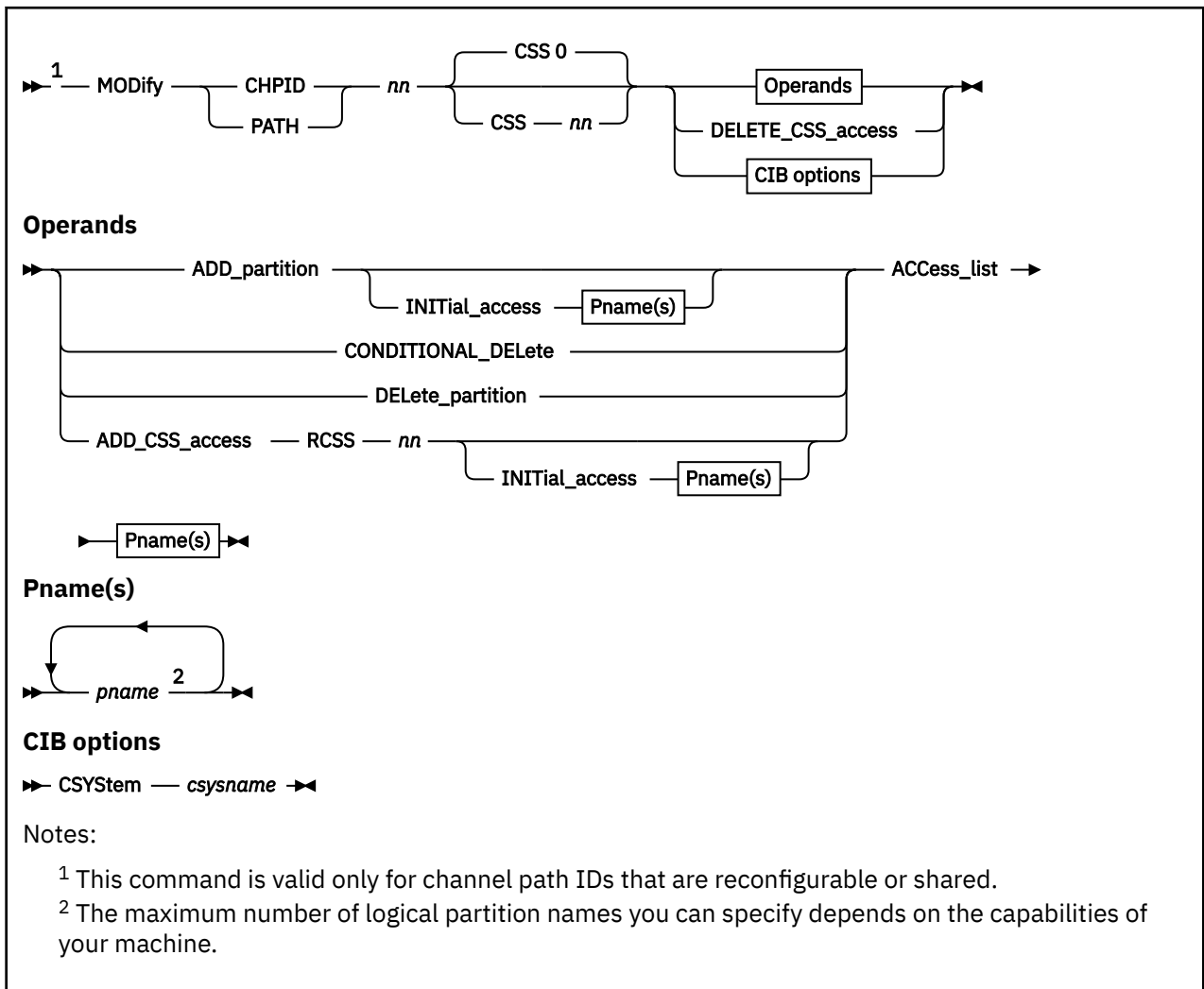
can be any character, number, or symbol.

## MESSAGE (User)

***msgtext***

identifies the message text.

# MODIFY CHPID / PATH



## Authorization

Privilege Class: B

## Purpose

Use MODIFY CHPID or PATH to change the definition of an existing reconfigurable or shared channel path identifier (CHPID) in the I/O configuration.

## Operands

**nn**

is the number of the channel path ID that you are changing. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**CSS nn**

is the channel-subsystem ID where the channel path is to be modified. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'. Your machine might support a lower maximum value.

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require that the logical channel-subsystem facility be installed.

**ADD\_partition**

tells CP that you want to add one or more logical partitions to the list of logical partitions that have access to this channel path in the specified or implied channel-subsystem image.

**CONDITIONAL\_DELeTe**

tells CP that you want to delete one or more logical partitions from the list of logical partitions that have access to this channel path in the specified or implied channel-subsystem image, but only if the channel path is not presently in use and configured to the logical partition.

**DELeTe\_partition**

tells CP that you want to delete one or more logical partitions from the list of logical partitions that have access to this channel path in the specified or implied channel-subsystem image, regardless of whether the channel path is in use or configured to one or more logical partitions.

**ADD\_CSS\_access**

tells CP that you want to add the channel path to the specified channel-subsystem image. See Usage Note “1” on page 680.

**RCSS nn**

is the reference channel-subsystem image from which the channel-path characteristics are to be copied. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'. Your machine might support a lower maximum value.

**INITIAL\_access pname**

specifies the name(s) of the logical partition(s) that will have initial access to this channel path in the specified or implied channel-subsystem image. The maximum number of logical partition names you can specify depends on the capabilities of your processor. Each *pname* must be a 1- to 8-digit character alphanumeric string.

**ACCess\_list pname**

specifies the name(s) of the logical partition(s) for which access to this channel path is being added or deleted in the specified or implied channel-subsystem image. The maximum number of logical partition names you can specify depends on the capabilities of your processor. If you are deleting the logical partitions from the access list, those logical partitions must already be in the access list for this channel path in the specified or implied channel-subsystem image. Each *pname* must be a 1- to 8-digit character alphanumeric string.

**DELETE\_CSS\_access**

tells CP that you want to delete the channel path from the specified the channel-subsystem image. See Usage Note “2” on page 680.

**CSYSstem csysname**

is the name of the target system to which the CIB CHPID connects. It is the name you would see after issuing the QUERY LSYSTEM command in a z/VM system running on that target machine. The name is established via an IOCP configuration definition, HCD configuration definition (in z/OS or z/VM), or the z/VM SET LSYSTEM operator command on the target machine. The variable *csysname* must be a 1- to 8-character alphanumeric string.

**Usage Notes**

1. On a machine with multiple logical channel-subsystem images, the DEFINE PATH and DEFINE CHPID commands first define the channel path in the single specified channel-subsystem image. If the channel path is a type that can span multiple logical channel-subsystem images, then MODIFY PATH or MODIFY CHPID commands must be entered for each additional channel-subsystem image in which you want to define the channel path.
2. On a machine with multiple logical channel-subsystem images, the DELETE CHPID and DELETE PATH commands are effective only for channel paths in a single channel-subsystem image. To delete channel paths that are in more than one channel-subsystem image, you must first enter MODIFY CHPID or MODIFY PATH commands to remove all but one channel-subsystem image. Then a DELETE CHPID or DELETE PATH command can be entered.



3. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the machine's active I/O configuration data set (IOCDS) file.

To change the machine's IOCDS file, you must:

- a. Update the IOCP source file.
  - b. Run the IOCP utility (see [“IOCP” on page 2203](#)) against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
  - c. Issue the SET IOCDS\_ACTIVE command (see [“SET IOCDS\\_ACTIVE” on page 1525](#)) to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR).
4. For more information about [Dynamically Changing Channel Paths \(CHPIDs\)](#) definitions in your I/O configuration, see *z/VM: I/O Configuration*.
  5. To define a new channel path ID on the system, use the DEFINE CHPID or PATH command. (See [“DEFINE CHPID / PATH” on page 132](#).)
  6. To delete an existing channel path ID on the system, use the DELETE CHPID or PATH command. (See [“DELETE CHPID / PATH” on page 259](#).)
  7. To change the definition of an existing nonreconfigurable channel path ID on the system, you must:
    - a. Delete the existing channel path definition, using the DELETE CHPID or PATH command.
    - b. Redefine the channel path definition, using the DEFINE CHPID or PATH command.
  8. To display the status of a specific logical channel path and the devices with that logical channel path installed or to display the type of a specific logical channel path, use the QUERY CHPID command. (See [“QUERY CHPID” on page 791](#).)
  9. To display the physical status of all 256 of the machine's channel paths, use the QUERY CHPIDS command. (See [“QUERY CHPIDS” on page 796](#).)
- Note:** If the channel path is in single channel service (SCS) mode, it will appear as "not configured" in the display.
10. To display the logical and physical status of the channel paths installed on one or more devices, use the QUERY PATHS command. (See [“QUERY PATHS” on page 1064](#).)

## Responses

Response 1:

To add 2 partitions to a reconfigurable channel path, enter the following:

```
modify chpid 1f add access_list vm6 vm9
Channel path configuration change completed successfully
Ready;
```

Response 2:

To delete 2 logical partitions from a reconfigurable channel path, enter the following:

```
modify chpid 1f conditional_delete access_list vm1 vm5
Channel path configuration change completed successfully
Ready;
```

Response 3:

To add access for an existing channel path to an additional channel-subsystem image, enter the following:

```
mod chpid 43 css 2 add_css rcss 1 init vm1 acc vm1 vm2 vm3
Channel path configuration change completed successfully
Ready;
```

Response 4:

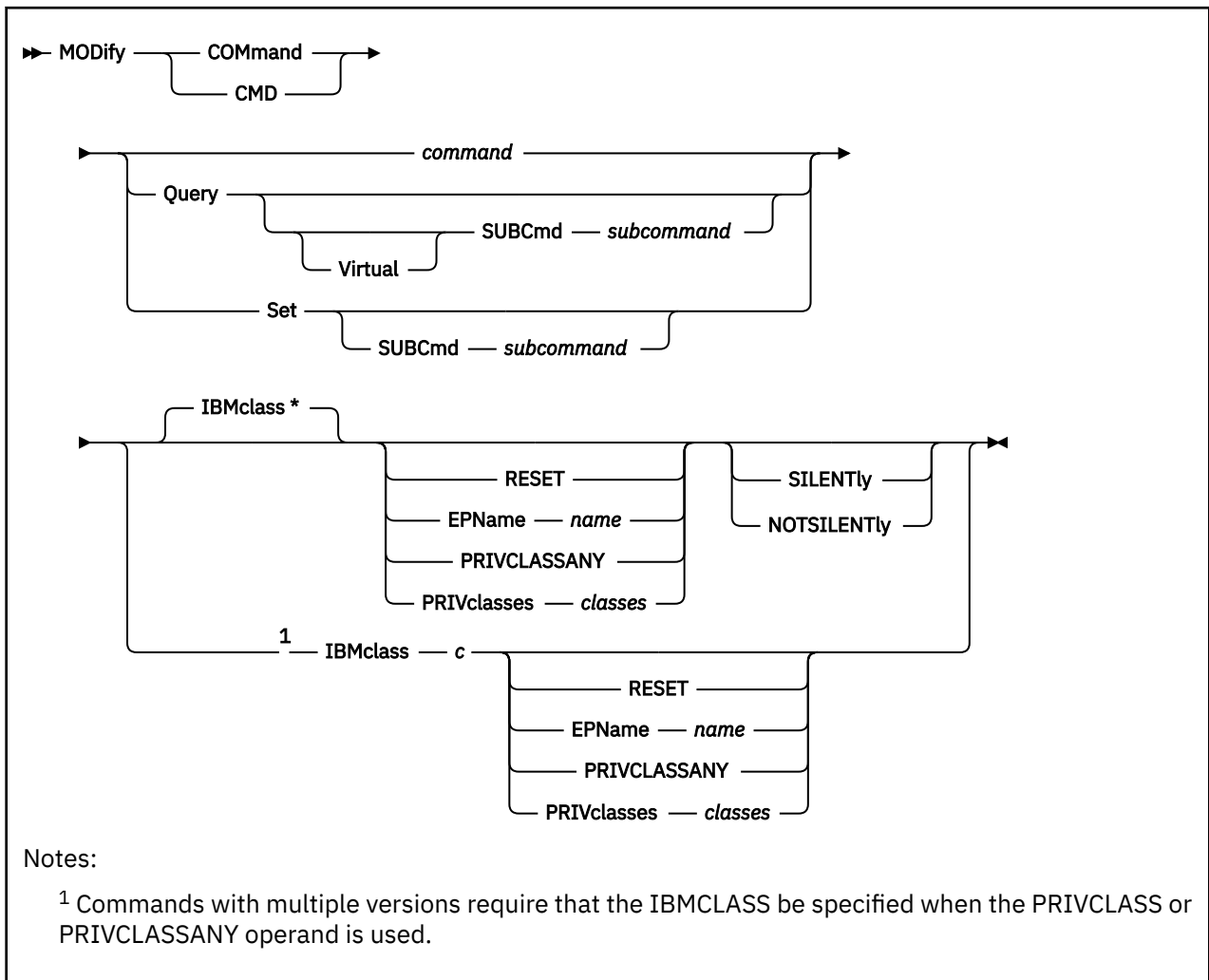
To change the name of the target system to which a CIB channel path connects, enter the following:

```
mod chpid 81 css 2 csystem machnum1
Channel path configuration change completed successfully
Ready;
```

### Messages

- HCP864E Partition *pname* does not have access to CSS *nn*
- HCP864E Partition *pname* not found
- HCP1006E Command processing cannot complete because the system does not support the coupling over InfiniBand channel-path facility.
- HCP1006E Command processing cannot complete because the system does not support the logical-channel-subsystem facility.
- HCP1006E The command processing cannot complete because the system is not running in EMIF mode
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request [for channel path|for control unit|for device] failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed {on this system|for CSS *nn*}
- HCP6819E A dynamic I/O command is currently executing for user *userid*

## MODIFY COMMAND / CMD



### Authorization

Privilege Class: A

### Purpose

Use `MODIFY COMMAND` or `MODIFY CMD` to redefine an existing CP command on the system. For more information about specifying generic command names, see Usage Note “1” on page 685.

You can also redefine an existing CP command during initialization using the `MODIFY COMMAND` or `MODIFY CMD` configuration file statements. For more information, see [MODIFY COMMAND / CMD Statement](#) in *z/VM: CP Planning and Administration*.

### Operands

#### *command*

is the name of the existing CP command that you are overriding. The variable *command* is a 1-character to 12-character alphanumeric string.

#### Query

tells CP that you are overriding a `QUERY` command.

**Virtual**

tells CP that you are overriding a QUERY VIRTUAL subcommand.

**SUBCmd *subcommand***

is the name of the QUERY subcommand that you are overriding. The variable *subcommand* is a 1- to 12-character alphanumeric string. For more information about specifying generic subcommand names, see Usage Note [“1” on page 685](#).

**Set**

tells CP that you are overriding a SET command.

**SUBCmd *subcommand***

is the name of the SET subcommand that you are overriding. The variable *subcommand* is a 1-character to 12-character alphanumeric string. For more information about specifying generic subcommand names, see Usage Note [“1” on page 685](#).

**IBMclass \***

tells CP to redefine all versions of the specified command or subcommand. If omitted, IBMCLASS \* is the default.

**IBMclass c**

tells CP to redefine a specific version of the specified command or subcommand. The variable *c* can be any 1 of the following:

**A**

this is a system-control command to be used by the primary system operator.

**B**

this is a command for operational control of real devices.

**C**

this is a command to alter system storage.

**D**

this is a command for system-wide control of spool files.

**E**

this is a command to examine system storage.

**F**

this is a command for service control of real devices.

**G**

this is a general-use command used to control the functions of a virtual machine.

**O**

(zero) this command has no specific IBM class assigned.

**RESET**

tells CP to restore the command to the originally defined settings.

**EPName *name***

tells CP the name of the entry point that contains the code to process the command. The variable *name* must be a 1-character to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

**PRIVCLASSANY**

tells CP that users with any privilege class can issue the command that you are redefining.

**PRIVclasses *classes***

tells CP that only users with one or more of the specified privilege classes can issue the command that you are redefining. Whatever you specify on this operand will replace the current privilege classes. The variable *classes* is 1 or more privilege classes in the range A through Z, 1 through 6, or an asterisk (\*). Privilege class \* indicates all privilege classes (A-Z and 1-6).

**Note:** If you want more than one privilege class, specify your classes in one string of characters. Do not separate the classes with blank spaces. For example, specify "privclasses abc123", not "privclasses a b c 1 2 3".

### SILENTly

tells CP that the responses from the command you are redefining can be suppressed by invoking it using the SILENTly command. For more information, see [“SILENTly” on page 1798](#).

**Note:** Response suppression is supported only for the ATTACH, DETACH, and GIVE commands.

### NOTSILENTly

tells CP that the responses from the command you are redefining cannot be suppressed.

## Usage Notes

1. If you are making similar changes to several QUERY (or SET) subcommands that share common characters, you can use a generic name rather than issuing multiple MODIFY COMMAND or CMD commands. A generic name is a 1- to 12-character alphanumeric string with asterisks (\*) in place of one or more characters and percent signs (%) in place of exactly one character. For example:

```
modify command query subcmd tr*c% ...
```

redefines all QUERY subcommands that start with TR and have C as their next-to-last character.

2. The subcommands that make up the set of QUERY subcommands are divided into two groups and are controlled by two separate MODIFY commands. For example, QUERY DASD is a subcommand and QUERY VIRTUAL DASD is a separate subcommand. You must specify the VIRTUAL operand on the MODIFY command if you wish to change the QUERY VIRTUAL DASD subcommand. Because of the additional granularity that is provided by the MODIFY command, you must issue two MODIFY commands if you wish to change both the virtual and non-virtual subcommands of QUERY. For example, to change all versions of the QUERY DASD and QUERY VIRTUAL DASD subcommands so that only user privilege class "X" can issue them, enter the following:

```
modify cmd query virtual subcmd DASD ibmc * priv x
modify cmd query subcmd DASD ibmc * priv x
```

The first MODIFY command changes the QUERY VIRTUAL subcommands while the second changes the QUERY subcommands.

3. Several commands in the system share generic command control blocks. For example, QUERY 000E (Class B version) shares the same generic command control block with QUERY L00E (Class B version). The determination of the actual command being issued is not made until the command handler gets control. The generic command blocks exist only for Class B and G commands. In addition, they exist on the virtual subcommand chain. In order to change these commands, you must specify the "VIRTUAL SUBCMD \*" and the appropriate IBMCLASS. For example, to add a user privilege Class X to the Class B version, you would enter the following:

```
modify cmd query virtual subcmd * ibmc b priv bx
```

For more information, refer to [Modifying a Command in z/VM: CP Planning and Administration](#).

4. To define a new CP command or a new version (by IBM class) of an existing CP command, use the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD” on page 143](#)).
5. To define a new alias for an existing CP command, use the DEFINE ALIAS command (see [“DEFINE ALIAS” on page 127](#)).
6. Once defined, an alias name cannot be used again. An alias cannot be modified or eliminated; it can only be disabled.
7. To load the command processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD” on page 111](#)).
8. To activate a CP command:

## MODIFY COMMAND / CMD

- While defining it, use the ENABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD” on page 143](#)).
  - After defining it, use the ENABLE COMMAND or CMD command (see [“ENABLE COMMAND / CMD” on page 460](#)).
9. To display the address of the CP command table entry block, the current IBM class, and the current privilege class for a specified CP command, use the LOCATE CMDBK command (see [“LOCATE CMDBK” on page 603](#)).
  10. To deactivate a CP command:
    - While defining it, use the DISABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD” on page 143](#)).
    - After defining it, use the DISABLE COMMAND or CMD command (see [“DISABLE COMMAND / CMD” on page 313](#)).
  11. To remove the command processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD” on page 117](#)).
  12. MODIFY COMMAND RESET will not allow a RESET to occur if the resulting command versions would have different EPNAMEs. This can occur if you change the EPNAME for a command then add a new version for that command. To avoid this restriction, you should add new versions of a command before changing the EPNAMEs associated with the command.
  13. The SILENTly and NOTSILENTly option must be applied to all versions of a command.
  14. For more information about user-defined commands, see [z/VM: CP Exit Customization](#).

## Responses

### Response 1:

To change the CP SHUTDOWN command from privilege class A to privilege class S, enter the following:

```
modify command shutdown privclasses s
Ready;
```

### Response 2:

To change the version of the CP SET PRIVCLASS command that allows users with any privilege class to issue the command to a version that allows only users with privilege class G and Z to issue the command, enter the following:

```
locate cmdbk set subcmd privc
Subcommand  CMDBK      Alias      IBMCLASS  Privilege
PRIVCLASS   12345678 *          C          C
PRIVCLASS   76543218 *          0          <ANY>
Ready;

modify cmd set subcmd privc ibmclass 0 privclasses gz
Ready;

locate cmdbk set subcmd privc
Subcommand  CMDBK      Alias      IBMCLASS  Privilege
PRIVCLASS   12345678 *          C          C
PRIVCLASS   76543218 *          0          GZ
----- 58281098 *          0          <ANY>
Ready;
```

Issuing the LOCATE CMDBK commands before and after your MODIFY COMMAND command is not necessary, but it shows you how the command definition has changed.

### Response 3:

To change all IBM class E QUERY subcommands to privilege class G, enter the following:

```
modify cmd query subcmd * ibmclass e privclasses g
Ready;
```

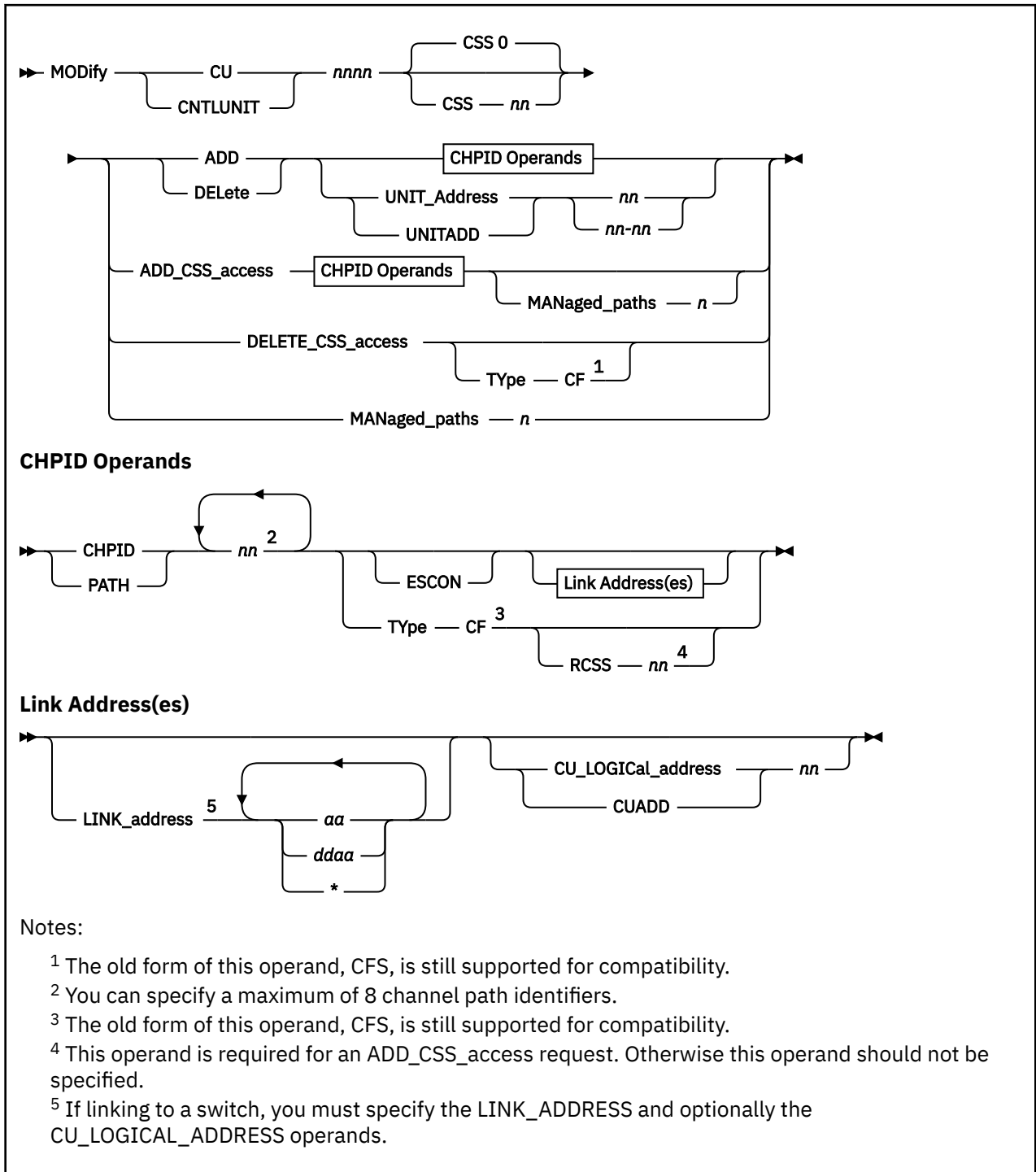
The preceding command does not change any QUERY VIRTUAL subcommands. To change them, enter the following:

```
modify cmd query virtual subcmd * ibmc e priv g
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP770E MODIFY is not valid for command {*command*|QUERY *subcommand*|SET *subcommand*}
- HCP770E MODIFY is not valid for command {*command*|QUERY *subcommand*|SET *subcommand*} because different EPNAMEs would be set
- HCP770E MODIFY RESET is not valid for command {*command*|QUERY *subcommand*|SET *subcommand*} because different EPNAMEs would be set
- HCP6704E Missing token at end of line
- HCP6706E Invalid command name - *command*
- HCP6706E Invalid entry point name - *epname*
- HCP6706E Invalid IBM class - *c*
- HCP6706E Invalid privilege classes - *classes*
- HCP6706E Invalid subcommand name - *subcommand*
- HCP6841E You did not specify an IBM class for command *command*
- HCP6842E Cannot find existing CP command
- HCP6845E You must specify a privilege class other than 0

## MODIFY CU / CNTLUNIT



### Authorization

Privilege Class: B

### Purpose

Use **MODIFY CU** or **CNTLUNIT** to change the definition of an existing control unit in the I/O configuration.



## Operands

### *nnnn*

is the number of the control unit that you are changing. The variable *nnnn* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFE'.

### **CSS** *nn*

is the channel-subsystem image in which the control unit is to be modified. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'.

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require that the logical channel-subsystem facility be installed.

### **ADD**

tells CP that you want to add one or more channel paths or unit addresses to this control unit.

### **DELeTe**

tells CP that you want to delete one or more existing channel paths or unit addresses from this control unit.

**Note:** You cannot use the DELETE operand to delete all of the existing channel paths or unit addresses from this control unit. To delete all of the existing channel paths, use the DELETE CHPID or PATH command. (See [“DELETE CHPID / PATH”](#) on page 259.) To delete all of the existing unit addresses, use the DELETE CU or CNTLUNIT command. (See [“DELETE CU / CNTLUNIT”](#) on page 263.)

### **ADD\_CSS\_access**

tells CP that you want to add the control unit to the specified channel-subsystem image. See Usage Note [“1”](#) on page 691.

### **DELETE\_CSS\_access**

tells CP that you want to delete the control unit from the specified the channel-subsystem image. See Usage Note [“2”](#) on page 691.

### **MANAged\_paths** *n*

tells CP that you want to modify the maximum number of managed channel paths which can be attached to the specified control unit or, for an ADD\_CSS\_access request, is the maximum number of managed channel paths in the specified channel-subsystem image that can be attached to the specified control unit. The variable *n* must be a 1-digit decimal number from 0 to 7.

**Note:** The number of MANAged\_paths plus the number of static paths attached to the specified control unit cannot exceed 8.

### **CHPID** *nn*

### **PATH** *nn*

tells CP which channel paths you want to add to or delete from this control unit or, for an ADD\_CSS\_access request, identifies the channel paths in the specified channel-subsystem image associated with this control unit. You can specify up to 8 channel path IDs. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**Note:** You cannot specify managed channel paths.

### **UNIT\_Address** *nn*

### **UNIT\_Address** *nn-nn*

### **UNITADD** *nn*

### **UNITADD** *nn-nn*

tells CP which unit addresses you want to add to or delete from this control unit. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

### **ESCON**

tells CP that this control unit uses the Enterprise Systems Connection (ESCON) channel to control unit interface to transmit data through optical cables. Also, specify ESCON for a control unit that connects to ESCON unswitched channels.

**LINK\_address aa**

**LINK\_address ddaa**

**LINK\_address \***

Each link-address value you specify identifies a port on a destination switch (ESCON or FICON director) to which the control unit is connected.

The LINK\_address operand must be specified when one or more of the channel paths to the control unit are switched paths; that is, when any of the paths traverse an ESCON or FICON director and for an ESCON director the path through the director is not dedicated. When all of the channel paths are unswitched (including dedicated paths through an ESCON director), the LINK\_address operand is optional.

For an unswitched FICON channel path, the end nodes determine the link addresses used at each end. Thus you cannot specify a numeric link address to be used. If you specify a LINK\_address operand value for an unswitched FICON channel path (for example, because other paths to the control unit are switched and a "placeholder" value is required), you must specify an asterisk (\*) as the link-address value.

When any link addresses are required to be specified (for example, when one or more of the channel paths is a switched path), you must provide a link address for each channel-path identifier (PATH or CHPID) that you specified, in the same order. When no link addresses are required to be specified (for example, when all of the channel paths are unswitched paths), the link address for each ESCON channel path is determined by the hardware processor, and the link address for each FICON channel path is negotiated between the hardware endpoints.

The link address is a one or two digit hexadecimal value for control units connected to an ESCON channel. The link address is a one, two, three, or four digit hexadecimal value for control units connected to a FICON channel. Leading zeroes can be omitted, except that an actual *dd* value of zero (see below) must be specified with at least one digit.

*aa* is the one-byte port-area address in an ESCON or FICON director to which the control unit is connected. This address must be a one or two digit hexadecimal number between X'01' and X'FE' for ESCON directors, and between X'00' and X'FF' for FICON directors.

Link addresses for channel paths having dedicated (unswitched) connections through an ESCON director can be arbitrary within the above-specified range, but it is recommended that the actual port-area address be used.

*ddaa* is the first two bytes of a standard three-byte fibre-channel port address. The third byte of a standard port address is used for fibre-channel arbitrated loops, which are not supported for FICON channels and therefore you do not specify. This two-byte form of the link address is required when the channel path is a FICON channel path that traverses more than one FICON director to reach any of the channel-path's destination control units.

This means that if the channel path traverses more than one FICON director to reach this control unit, a two-byte link address is required. In addition, if the channel path traverses only one FICON director to reach this control unit, but the same channel path traverses more than one FICON director to reach some other control unit, a two-byte link address is required for that channel path in all control-unit definitions in which the channel path appears. That is, once a two-byte link address is required for any control unit on a channel path, a two-byte link address is required for all control units on that channel path.

*dd* is the switch address (or domain identifier) of the destination FICON director (for example., the switch to which the control unit is directly connected), and *aa* is the port-area address within that FICON director. The link address must be a three to four digit hexadecimal number between X'0000' and X'FFFF'.

One-byte and two-byte link addresses can be mixed within the LINK\_address operand, according to the requirements for each channel path. You can choose to use a two byte link address when only a one byte address is required as long as the FICON director is configured with the high-integrity attributes (see the *IOCP User's Guide* for more information).

**CU\_LOGICAL\_address nn****CUADD**

is the logical address of the control unit that you are changing. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FE'. If omitted, the default is X'00'.

**Type CF**

tells CP that this control unit is connected to coupling facility channel paths. Coupling facility channel paths include coupling facility sender (CFS), coupling facility receiver (CFR), coupling facility peer (CFP), cluster bus sender (CBS), cluster bus receiver (CBR), cluster bus peer (CBP), or internal coupling peer (ICP) channel paths.

**RCSS nn**

is the reference channel-subsystem image from which the control unit attributes are to be copied for the specified coupling-facility control unit that is being added to the additional channel-subsystem image specified by the *CSS nn* parameter. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'.

**Usage Notes**

1. On a machine with multiple logical channel-subsystem images, the DEFINE CU and DEFINE CNTLUNIT commands first define the control unit on channel paths within the single specified channel-subsystem image. If the control unit is connected to channel paths in more than one channel-subsystem image, then MODIFY CU or MODIFY CNTLUNIT commands must be entered to define the control unit's connections to each group of channel paths in the additional channel-subsystem images.
2. On a machine with multiple logical channel-subsystem images, the DELETE CU and DELETE CNTLUNIT commands are effective only for control units in a single channel-subsystem image. To delete control units that are in more than one channel subsystem image, you must first enter MODIFY CU or MODIFY CNTLUNIT commands to remove all but one channel-subsystem image. Then a DELETE CU or DELETE CNTLUNIT command can be entered.
3. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.  
To change the IOCDS file on the SE hard disk, you must:
  - a. Update the IOCP source file.
  - b. Run the IOCP utility (see “IOCP” on page 2203) against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
  - c. Issue the SET IOCDS\_ACTIVE command (see “SET IOCDS\_ACTIVE” on page 1525) to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR).
4. For more information about changing control unit definitions in your I/O configuration, see [Dynamically Changing Channel Paths \(CHPIDs\)](#) in *z/VM: I/O Configuration*.
5. To define a new control unit on the system, use the DEFINE CU or CNTLUNIT command. (See “DEFINE CU / CNTLUNIT” on page 161.)
6. To delete an existing control unit on the system, use the DELETE CU or CNTLUNIT command. (See “DELETE CU / CNTLUNIT” on page 263.)

**Responses**

Response 1:

To add unit addresses X'01' through X'1F' to existing control unit 1200, enter the following:

```
modify cu 1200 add unit_address 01-1f
Control unit configuration change completed successfully
Ready;
```

Response 2:

To add a link address of X'F4' and a control unit logical address of X'00' to channel path 19 on control unit 2F00, enter the following:

```
mod cu 2f00 add chpid 19 escon link_addr f4 cu_logical_addr 00
Control unit configuration change completed successfully
Ready;
```

Response 3:

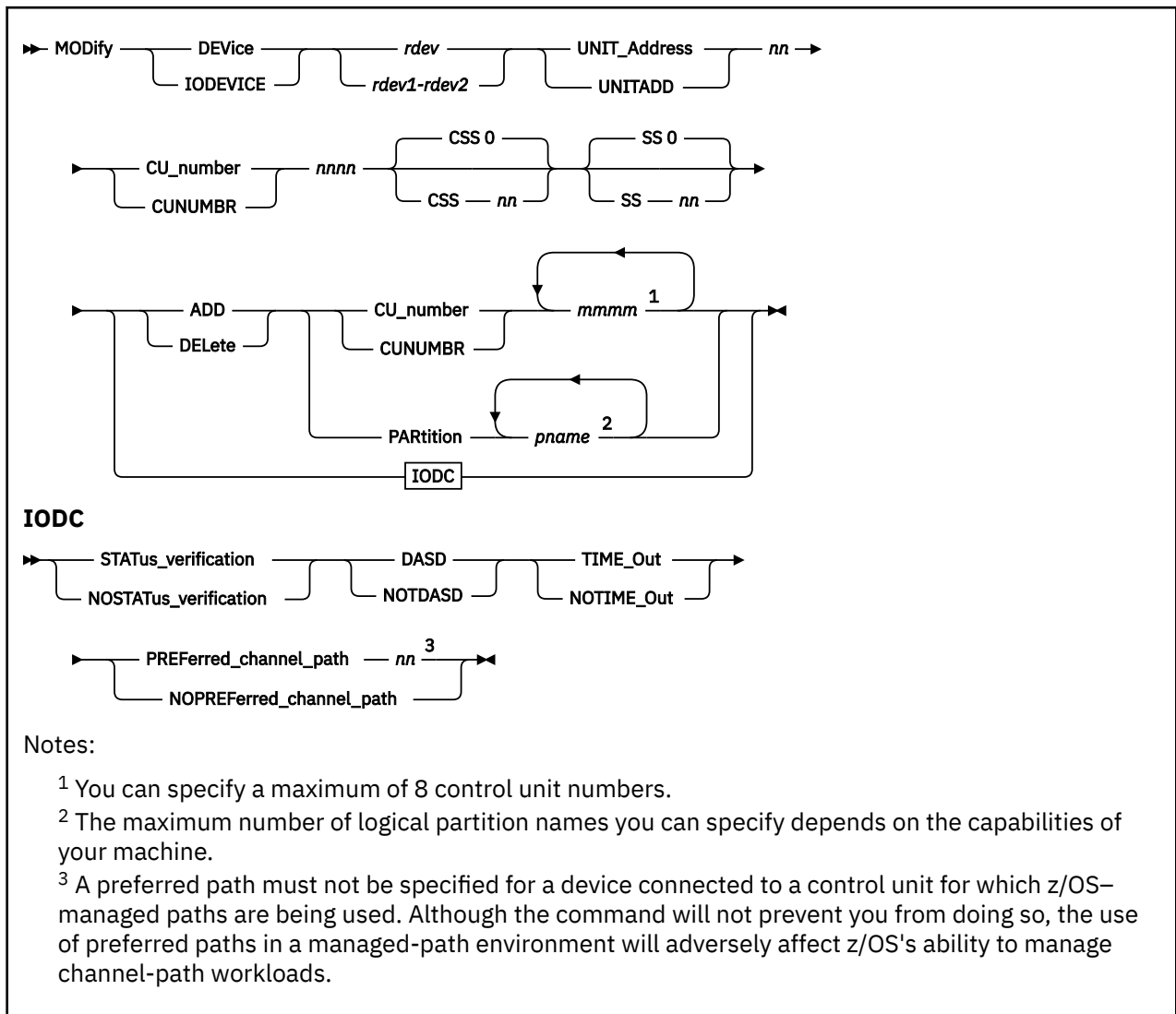
To add access to existing control unit 1210 from an additional channel-subsystem image, enter the following:

```
mod cu 1210 css 3 add_css chpid 9 link 43 cuadd 81 man 2
Control unit configuration change completed successfully
Ready;
```

### Messages

- HCP1006E Command processing cannot complete because the system does not support the logical-channel-subsystem facility.
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request [for channel path|for control unit|for device] failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed {on this system|for CSS *nn*}
- HCP6818E Modify command failed. Channel path *xx* is logically online
- HCP6819E A dynamic I/O command is currently executing for user *userid*
- HCP6854E CU *cu* has been created but is in definition error. Devices on the CU are not operational and cannot be varied online
- HCP6536E The link address *nn* exceeds two digits
- HCP6536E Number of link addresses does not match the number of channel paths

## MODIFY DEVICE / IODEVICE



### Authorization

Privilege Class: B

### Purpose

Use **MODIFY DEVICE** or **IODEVICE** to change the definition of one or more existing real devices in the I/O configuration.

**Note:** You cannot modify an open systems adapter (OSA) device. If you need to change the definition of an OSA device, you must delete the device and define it again with the new information.

### Operands

*rdev*

*rdev1-rdev2*

is the real device number (or range of real device numbers) of the device that you are changing.

**UNIT\_Address *nn*****UNITADD *nn***

is the last 2 digits of the device address or number. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**Note:**

1. If you are specifying a range of devices (*rdev1-rdev2*), the unit address that you specify is the starting address associated with the first device in the range.

**CU\_number *nnnn*****CUNUMBR *nnnn***

is the number of one of the control units associated with the device that you are changing. The variable *nnnn* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFE'.

**CSS *nn***

is the channel-subsystem image in which the devices are to be modified. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'.

**Note:** This parameter and other parameters that identify different channel-subsystem IDs require that the logical channel-subsystem facility be installed.

**SS *nn***

is the subchannel set with which this device is associated. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'03'.

**Note:** This parameter requires that the multiple-subchannel-set facility be installed.

**ADD**

tells CP that you are adding one or more control units or logical partition accesses to this device.

**DELeTe**

tells CP that you are deleting one or more control units or logical partition accesses from this device.

**CU\_number *mmmm*****CUNUMBR *mmmm***

tells CP which control units you are adding to or deleting from this device. You can specify up to 8 control unit numbers. Each *mmmm* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFE'.

**PARTition *pname***

tells CP the logical partitions whose access you are adding to or deleting from this device. The maximum number of logical partition names you can specify depends on the capabilities of your machine. Each *pname* must be a 1- to 8-character alphanumeric string.

**STATus\_verification**

tells CP that the status verification facility is active for this device, causing the channel subsystem to check the device's status every time this device announces its status.

**NOSTATus\_verification**

tells CP that the status verification facility is not active for this device, and so the channel subsystem will not check the device's status.

**DASD**

tells CP that this device is a direct access storage device (DASD).

**NOTDASD**

tells CP that this device is not a direct access storage device (DASD).

**TIME\_Out**

tells CP that the I/O interface time out function is active for the following I/O interfaces between the channel and I/O device:

- 4 second time out for initial selection
- 30 second time out for data transfer.

**NOTIME\_Out**

tells CP that the I/O interface time out function is not active for this device.

**PREFerred\_channel\_path nn**

tells CP to use this channel path in the specified or implied channel-subsystem image to initiate I/O requests before trying any other channel paths from the device that you are changing. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

**Note:** A preferred path must not be specified for a device connected to a control unit for which z/OS-managed paths are being used. Although the command will not prevent you from doing so, the use of preferred paths in a managed-path environment will adversely affect z/OS's ability to manage channel-path workloads.

**NOPREFerred\_channel\_path**

tells CP that there is no preferred channel path.

**Usage Notes**

1. If you are using this command to modify I/O device characteristics, then you must specify one of each of the I/O device characteristics (IODC), because this command does not accept defaults and will not carry over any previous IODC characteristics definitions.

For example, suppose you used the following command to define a new DASD:

```
define device 16d9 unit_address a3 cu 501 dasd
```

In this command, you specified one IODC characteristic, DASD. You took the defaults for the other three: no preferred channel path, no status verification, and no time out function.

If you want to change the DASD to another device and retain the previous IODC characteristic definitions, you must specify exactly what you want:

```
mod dev 16d9 unit_addr a3 cu 501 nostat notdasd notime_out nopref
```

**Note:** This is especially true for DASD devices. If you are modifying the IODC characteristics of a DASD, you must specify DASD in your command. If you do not, the DASD will not work as expected.

2. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.
  - b. Run the IOCP utility (see [“IOCP” on page 2203](#)) against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
  - c. Issue the SET IOCDS\_ACTIVE command (see [“SET IOCDS\\_ACTIVE” on page 1525](#)) to make the new IOCDS file the active file that the machine will use during the next power-on reset (POR).
3. For more information about changing real device definitions in your I/O configuration, see [I/O Configuration Definition Process in z/VM: I/O Configuration](#).
  4. To define a new device on the system, use the DEFINE DEVICE or IODEVICE command. (See [“DEFINE DEVICE / IODEVICE” on page 167](#).)
  5. To delete an existing device on the system, use the DELETE DEVICE or IODEVICE command. (See [“DELETE DEVICE / IODEVICE” on page 265](#).)
  6. To display the status of one or more real devices, use the QUERY (Real Device) command. (See [“QUERY \(Real Device\)” on page 1114](#).)

**Note:** The QUERY (Real Device) command displays the status of the device only if you issue the QUERY command in a logical partition (LPAR) that has been defined to have access to the device.

**Responses**

Response 1:

To change the IO DC characteristics of device 300, enter the following:

```
mod dev 300 unit_addr 0 cu 300 nostatus dasd notime_out nopreferred
Device configuration change completed successfully
Ready;
```

Response 2:

To add another control unit to the 32 devices at addresses X'3200' through X'321F', enter the following:

```
modify dev 3200-321f unit_addr 20 cu 1200 add cu 1300
Device configuration change completed successfully
Ready;
```

Response 3:

To delete a logical partition from the 64 devices at addresses X'4640' through X'467F', enter the following:

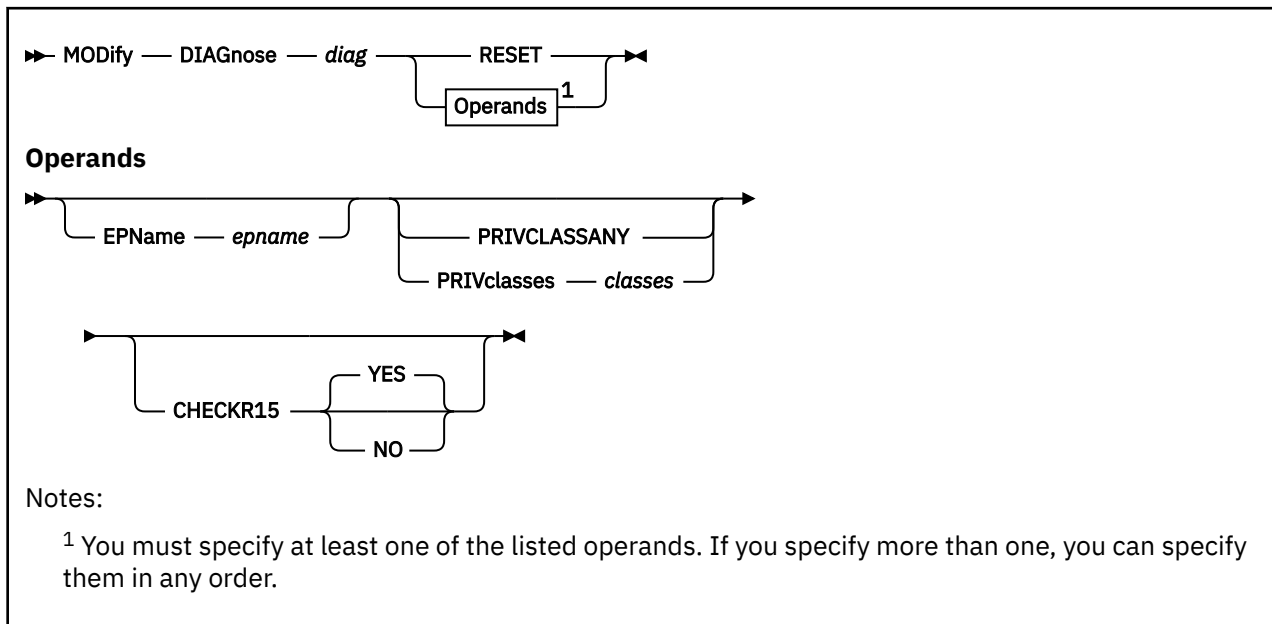
```
mod dev 4640-467F unit_addr 40 cu 4000 delete partition cambridg
Device configuration change completed successfully
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP864E Partition *pname* does not have access to CSS *nn*
- HCP864E Partition *pname* not found
- HCP1006E Command processing cannot complete because the system does not support the logical-channel-subsystem facility.
- HCP1006E The command processing cannot complete because the system is not running in EMIF mode
- HCP1006E Command processing cannot complete because the system does not support the multiple-subchannel-set facility.
- HCP6805E Cannot determine partition being changed by this dynamic I/O request
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request [for channel path|for control unit|for device] failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed {on this system|for device *rdev*|for CSS *nn*}
- HCP6818E Modify command failed. Channel path *path* is logically online
- HCP6818E Modify command failed. Device *rdev* is a base Parallel Access Volume for which an alias exists
- HCP6818E Modify command failed. Device *rdev* is not subchannel offline
- HCP6819E A dynamic I/O command is currently executing for user *userid*.



## MODIFY DIAGNOSE



### Authorization

Privilege Class: A

### Purpose

Use MODIFY DIAGNOSE to redefine an existing DIAGNOSE code on the system.

You can also redefine an existing DIAGNOSE code during initialization using the `MODIFY DIAGNOSE` configuration file statement. For more information, see *z/VM: CP Planning and Administration*.

### Operands

#### *diag*

is the number of the DIAGNOSE code that you are redefining.

#### RESET

tells CP to stop using the customer-written DIAGNOSE code and return to using the existing DIAGNOSE code that was shipped with the z/VM product.

#### CHECKR15 YES

indicates that the diagnose router should check register 15 upon return from the diagnose handler.

#### CHECKR15 NO

indicates that the diagnose router should not check register 15 upon return from the diagnose handler.

#### EPName *name*

tells CP the name of the entry point that contains the code to process the DIAGNOSE code. The variable *name* must be a 1- to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

#### PRIVCLASSANY

tells CP that users with any privilege class can issue the DIAGNOSE code that you are redefining.

**PRIVclasses classes**

tells CP that only users with 1 or more of the specified privilege classes can issue the DIAGNOSE code that you are redefining. Whatever you specify on this operand will replace the current privilege classes. The variable *classes* is 1 or more privilege classes in the range A through Z, 1 through 6, or an asterisk (\*). Privilege class \* indicates all privilege classes (A-Z and 1-6).

**Note:** If you want more than one privilege class, specify your classes in one string of characters. Do not separate the classes with blank spaces. For example, specify "privclasses abc123", not "privclasses a b c 1 2 3".

**Usage Notes**

1. To define a new DIAGNOSE code, use the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE”](#) on page 172).
2. To load the DIAGNOSE processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD”](#) on page 111).
3. To activate a new DIAGNOSE code:
  - While defining it, use the ENABLE operand of the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE”](#) on page 172).
  - After defining it, use the ENABLE DIAGNOSE command (see [“ENABLE DIAGNOSE”](#) on page 463).
4. To display information about a DIAGNOSE code (status, entry point name, and privilege class), use the QUERY DIAGNOSE command (see [“QUERY DIAGNOSE”](#) on page 885).
5. To display the address of the CP DIAGNOSE code table block for a DIAGNOSE code, use the LOCATE DGNBK command (see [“LOCATE DGNBK”](#) on page 608).
6. To deactivate a DIAGNOSE code:
  - While defining it, use the DISABLE operand of the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE”](#) on page 172).
  - After defining it, use the DISABLE DIAGNOSE command (see [“DISABLE DIAGNOSE”](#) on page 316).
7. To remove the DIAGNOSE processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117).
8. Many external security managers (ESMs) do not support DIAGNOSE codes above X'03FC'. For this reason, CP does not support DIAGNOSE codes above X'03FC'. The DIAGNOSE codes between X'0000' and X'03FC' are divided as follows:
  - X'0000' to X'00FC'**  
Reserved for IBM use
  - X'0100' to X'01FC'**  
Reserved for customer use
  - X'0200' to X'03FC'**  
Reserved for IBM use.
9. When CHECKR15 YES is specified, the diagnose router will check register 15 on return from the diagnose handler. If register 15 contains:
  - RC = 0**  
Processing was successful. Complete the guest instruction.
  - RC = 4**  
Processing failed due to a condition which would cause a guest program check. Simulate guest program interruption passed in R0.
  - RC = 8**  
Nullify the instruction.
  - RC = 12**  
Present the machine check then nullify the instruction. R2 will contain the address of the MCRBK which will contain the machine check information.

**RC = 16**

Generate machine check for processing damage, then go to HCPENDOP to terminate the instruction.

**RC = 20**

Present the machine check, then go to HCPENDOP to terminate the instruction. R2 will contain the address of the MCRBK, which contains machine check information.

**RC = 24**

Issue error message or soft abend for paging I/O error, then nullify the instruction. R1 has the message or abend number.

If a return code is invalid (negative, not a multiple of 4 or too big ( RC > 24 )), then a soft abend will occur.

10. For more information about user-defined DIAGNOSE codes, see [Creating, Controlling, and Redefining CP Commands and Diagnose Codes in z/VM: CP Exit Customization](#).

## Responses

Response 1:

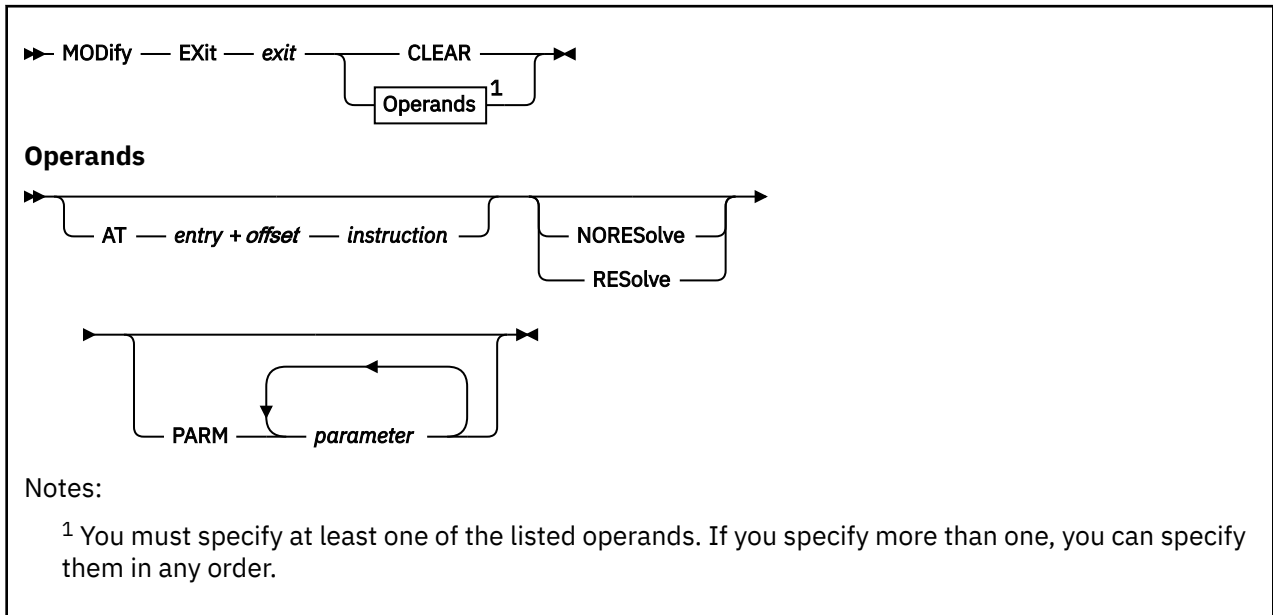
To change the privilege class for DIAGNOSE code X'7C' from "any" to R, enter the following:

```
modify diagnose 7c privclasses r
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP770E MODIFY is not valid for Diagnose code *diag*
- HCP2751E Diagnose code *diag* does not exist
- HCP2751E Diagnose codes *diag1-dia2* do not exist
- HCP6704E Missing token at end of line
- HCP6706E Invalid Diagnose code - *diag*
- HCP6706E Invalid entry point name - *eptime*
- HCP6706E Invalid privilege classes - *classes*
- HCP6845E You must specify a privilege class other than 0

## MODIFY EXIT



### Authorization

Privilege Class: A

### Purpose

Use MODIFY EXIT to redefine or remove an existing dynamic exit point.

You can also redefine or remove an existing dynamic exit point during system initialization using the MODIFY EXIT configuration file statement. For more information, see [MODIFY EXIT Statement](#) in *z/VM: CP Planning and Administration*.

### Operands

#### exit

is the number of the exit point you are redefining or removing. This value must be a hexadecimal number between X'0' and X'FFFF'. The recommended value is between X'F000' and X'FFFF', because that range is reserved for customer use. All other exit numbers are reserved for IBM, vendor, or general use.

#### CLEAR

tells CP to remove the exit.

#### AT entry + offset instruction

identifies the new location for the exit point you are redefining and the instruction that is located at the exit point. The variable *entry* must be a 1-to-8-character string. The first character must be alphabetic or one of the following special characters: \$ (dollar sign), # (number sign), \_ (underscore), or @ (at sign). The rest of the string can be alphabetic or numeric characters, the four special characters (\$, #, \_, or @), or any combination. The variable *offset* must be a 1-to-4-character even hexadecimal number between X'0' and X'FFFE'. The variable *instruction* must be a 2-, 4-, or 6-character hexadecimal number.

#### NORESolve

tells CP to resolve the entry points associated with this exit number the first time they are called.

**RESolve**

tells CP to resolve the entry points associated with this exit number when the association is first established. Any existing associated entry points are resolved immediately.

**PARM parameter**

is a list of one or more parameters to be supplied to the exit. Five kinds of tokens can be used to define a parameter:

1. Addresses: strings up to eight characters long, consisting of the hexadecimal digits 0 through 9 and A through F.
2. General Registers: strings beginning with G or R, followed by a decimal number between 0 and 15 or a hexadecimal digit, designating the contents of a general register.
3. Indirection: a percent sign (%), which causes the contents of an address or the contents of an address in a register to be used instead of the address or register contents itself.
4. Arithmetic: a plus sign (+) or minus sign (-).
5. Displacement: strings of up to four hexadecimal digits.

Each parameter string specifies how to combine these tokens to generate a parameter value to be passed to an exit routine. The following is a Backus-Naur definition of the syntax of a parameter:

```
<parameter> ::= <anchor> | <anchor><vector>
<anchor> ::= <reg> | 0..FFFFFFFF | <anchor>%
<vector> ::= <modifier> | <vector>% | <vector><modifier>
<modifier> ::= +<disp> | -<disp> | +<reg> | -<reg>
<reg> ::= G<digit> | R<digit>
<digit> ::= 0..15 | 0..9, A..F
<disp> ::= 0..7FFF
```

**Usage Notes**

1. To define a new dynamic exit point, use the DEFINE EXIT command (see [“DEFINE EXIT” on page 176](#)).
2. To load the exit processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD” on page 111](#)).
3. To activate a new exit point, use the ENABLE EXITS command (see [“ENABLE EXITS” on page 465](#)).
4. To display information about an exit point (status, entry point name, and parameters), use the QUERY EXITS command (see [“QUERY EXITS” on page 907](#)).
5. To deactivate an exit point, use the DISABLE EXITS command (see [“DISABLE EXITS” on page 319](#)).
6. To remove the exit processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD” on page 117](#)).
7. Exit numbers are allocated as follows:
  - X'0000' to X'7FFF' are reserved for IBM use.
  - X'8000' to X'FFFF' are reserved for vendor and general use.
  - X'F000' to X'FFFF' are reserved for private customer use.
8. The RESOLVE option ensures that the entry names associated with an exit point are defined.
9. Each exit is passed a parameter list that begins with three standard parameters, see [Usage Conventions in z/VM: CP Exit Customization](#). Additional parameters, specified by the PARM operand, are optional and follow the first three in the order in which they are specified.
10. Errors (for example, addressing exceptions) during evaluation of user-defined parameters when an exit is being invoked cause CP to abend.
11. For more information about user-defined exits, see [Dynamic Exit Points](#) in *z/VM: CP Exit Customization*.

### Examples

Example 1:

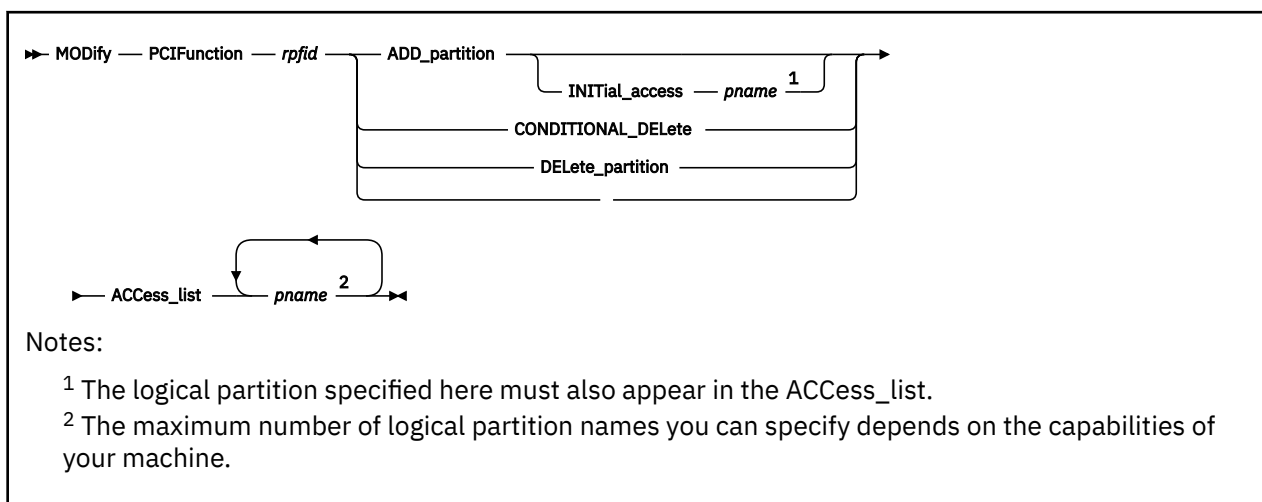
To change the location of exit point X'F422' to HCPXGMSM + 4, enter the following:

```
modify exit f422 at hcpxmgsM + 4 41700001  
Ready;
```

### Messages

- HCP002E Invalid operand — *operand*
- HCP003E Invalid option — command contains extra option(s) starting with *option*
- HCP005E Invalid hexdata — *hexdata*
- HCP010E Invalid register — *register*
- HCP013E Conflicting option — *option*
- HCP026E Operand missing or invalid
- HCP349E String supplied does not match storage contents
- HCP372E Invalid location for exit; command rejected
- HCP373E Exit already defined at location; command rejected
- HCP374E Exit number *exit* is not dynamic
- HCP375E Exit number *exit* is enabled or associated
- HCP376E Unknown entry point name *entry* cannot be associated with an exit point requiring resolution
- HCP1003E *module* was not found
- HCP2752E Exit number *exit* does not exist
- HCP6704E Missing token at end of line
- HCP6706E Invalid CP Exit number — *exit*
- HCP6706E Invalid instruction — *instruction*
- HCP6706E Invalid routine name — *routine*

## MODIFY PCIFUNCTION



### Authorization

Privilege Class: B

### Purpose

Use MODIFY PCIFUNCTION to modify a real PCI function in the I/O configuration.

### Operands

#### *rpfid*

is the real function identifier of the PCI function you are modifying. The *rpfid* must be a 1- to 8-digit hexadecimal number between X'00000000' and X'7FFFFFFF'. The maximum value of the *rpfid* is model dependent and your machine likely supports a lower maximum value.

#### **ADD\_partition**

tells CP that you want to add one or more logical partitions to the list of logical partitions that have access to this PCI function.

#### **CONDITIONAL\_DELETE**

tells CP that you want to delete one or more logical partitions from the list of logical partitions that have access to this PCI function, but only if the PCI function is not presently configured to any logical partition in the I/O configuration.

#### **DELETE\_partition**

tells CP that you want to delete one or more logical partitions from the list of logical partitions that have access to this PCI function regardless of whether the PCI function is presently configured to any logical partition in the I/O configuration.

#### **ACCESS\_list pname**

specifies the name(s) of the logical partition(s) for which access to this PCI function is being added or deleted. The maximum number of logical partition names you can specify depends on the capabilities of your machine. Each *pname* must be a 1- to 8-digit character alphanumeric string.

#### **INITIAL\_access pname**

specifies the name of the logical partition that will have initial access to this PCI function.

**Note:** When a logical partition is specified with the INITIAL\_access parameter, it must also be specified as part of the ACCESS\_list.

### Usage Notes

1. After successfully entering this command, you will have changed the software's view of the I/O configuration and part of the hardware's view: the active I/O configuration in the hardware system area (HSA). You should also change the other part of the hardware's view: the active I/O configuration data set (IOCDS) file on the support element (SE) hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.
  - b. Run the IOCP utility against the updated IOCP file. If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
  - c. Issue the SET IOCDS\_ACTIVE command to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR).
2. For more information about changing real PCI functions in your I/O configuration, see [z/VM: I/O Configuration](#).
  3. To define a PCI function to the system, use the DEFINE PCIFUNCTION command.
  4. To delete an existing PCI function on the system, use the DELETE PCIFUNCTION command.
  5. To display the status of one or more real PCI functions, use the QUERY PCIFUNCTION command.

### Responses

Response 1:

To add access to a PCI function for two partitions, enter the following:

```
modify pcif 43 add init vm6 acc vm6 vm9
PCI function configuration change completed successfully
Ready;
```

Response 2:

To delete access from a PCI function for two partitions, enter the following:

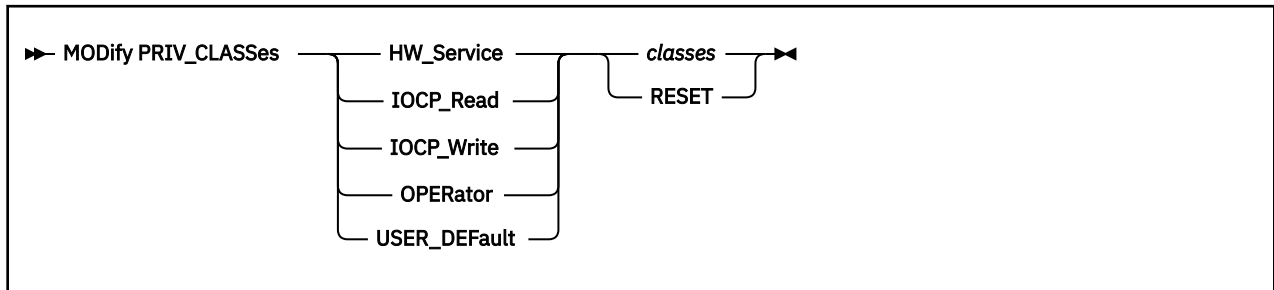
```
modify pcif 43 del acc vm3 vm5
PCI function configuration change completed successfully
Ready;
```

### Messages

- HCP864E Partition *pname* not found
- HCP1006E Command processing cannot complete because the system does not support the Dynamic-I/O-for-I/O-adapters facility.
- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request for PCI function failed with return code *nnnn*
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6818E Modify command failed. PCI function *rpfid* is not offline.
- HCP6819E A dynamic I/O command is currently executing for user *userid*



## MODIFY PRIV\_CLASSES



### Authorization

Privilege Class: A

### Purpose

Use MODIFY PRIV\_CLASSES to change the privilege classes authorizing the following CP functions:

- Logging on as the primary system operator
- Intensive error recording
- Using the read function of the CP IOCP utility
- Using the write function of the CP IOCP utility
- Specifying the default user class.

### Operands

#### HW\_Service

tells CP to change the privilege classes authorized to perform intensive error recording.

#### IOCP\_Read

tells CP to change the privilege classes authorized to use the read function of the IOCP utility. (For more information about the IOCP utility, see *z/VM: CP Planning and Administration*.)

#### IOCP\_Write

tells CP to change the privilege classes authorized to use the write function of the IOCP utility. (For more information about the IOCP utility, see *z/VM: CP Planning and Administration*.)

#### OPERator

tells CP to change the privilege classes which are for the primary system operator.

#### USER\_DEFault

tells CP to change the privilege classes that are defined for users who do not have a class specified in their directory entries.

#### RESET

tells CP to reset the privilege classes to their original default settings.

#### classes

tells CP which class or classes are to be set. The variable *classes* is a 1- to 8-character alphanumeric string from A through Z and from 0 to 6.

### Usage Notes

1. The configuration file statement PRIV\_CLASSES can be used to establish initial values.

### Responses

Response 1:

## MODIFY PRIV\_CLASSES

To authorize:

- The primary system operator for class A, C, and E commands
- Class C users to issue the IOCP READ and WRITE
- Class D users to perform intensive error recording
- All users without classes defined in their directory entries for class G commands

use the following MODIFY PRIV\_CLASSES commands:

```
MODify PRIV_CLASSES Operator ACE
Ready;
MODify PRIV_CLASSES IOCP_Write C
Ready;
MODify PRIV_CLASSES IOCP_Read C
Ready;
MODify PRIV_CLASSES HW_Service D
Ready;
MODify PRIV_CLASSES User_Default G
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP770E Modify RESET is not valid for PRIV\_CLASSES *type*
- HCP6704E Missing token at end of line
- HCP6706E Invalid privilege classes - *class*

# MONITOR

## Purpose

Use the MONITOR commands to control the selection, collection, and reporting of data from the host system:

- MONITOR EVENT controls monitoring of event data
- MONITOR SAMPLE controls monitoring of sample data
- MONITOR START starts both event and sample monitoring
- MONITOR STOP stops all monitoring.

## Usage Notes

(General for MONITOR)

1. MONITOR commands can be run from virtual machines with the A/E privilege. In order to avoid conflict among the commands, the administrator should be aware of the virtual machines that have the privilege to run the monitor commands.
2. MONITOR START can be issued before or after a user connects to \*MONITOR. If the MONITOR START command is issued before any connection to \*MONITOR, monitor recording goes into a pending state until a user connects to \*MONITOR. The following must be done before the monitor recording gets activated:
  - A saved segment must be defined and saved for the monitor
  - An application program running in a virtual machine must load the monitor saved segment and connect to the \*MONITOR CP system service
  - A MONITOR START command must be issued.
3. Both the event and sample profiles can be changed after monitoring has started. Note that:
  - Partition size of the saved segment cannot be changed unless event monitoring is stopped.

New event domains are activated upon completion of the EVENT command, and new sample domains are activated when the next interval elapses.
4. For a detailed description of data collected, refer to the monitor records in [z/VM: Performance](#).
5. The valid classes and types for use with Seeks and I/O domains are listed in the following table:

CLASS	TYPE	Other TYPES Included	Valid for Seeks	Valid for I/O
	3480			X
	3490			X
	3590			X
	3380		X	X
	3390		X	X
	9336		X	X
	FB-512 <sup>1</sup>	9336	X	X
SPOOL	1403			X
	3203			X

Table 31. Valid Classes and Types for Seeks and I/O Domains (continued)				
CLASS	TYPE	Other TYPES Included	Valid for Seeks	Valid for I/O
	3262			X
	3505			X
	3525			X
	3800			X
	4245			X
	4248			X
GRAF	GPRT <sup>2</sup>			X
	GTERM <sup>2</sup>			X
PCIF				X

**Note:**

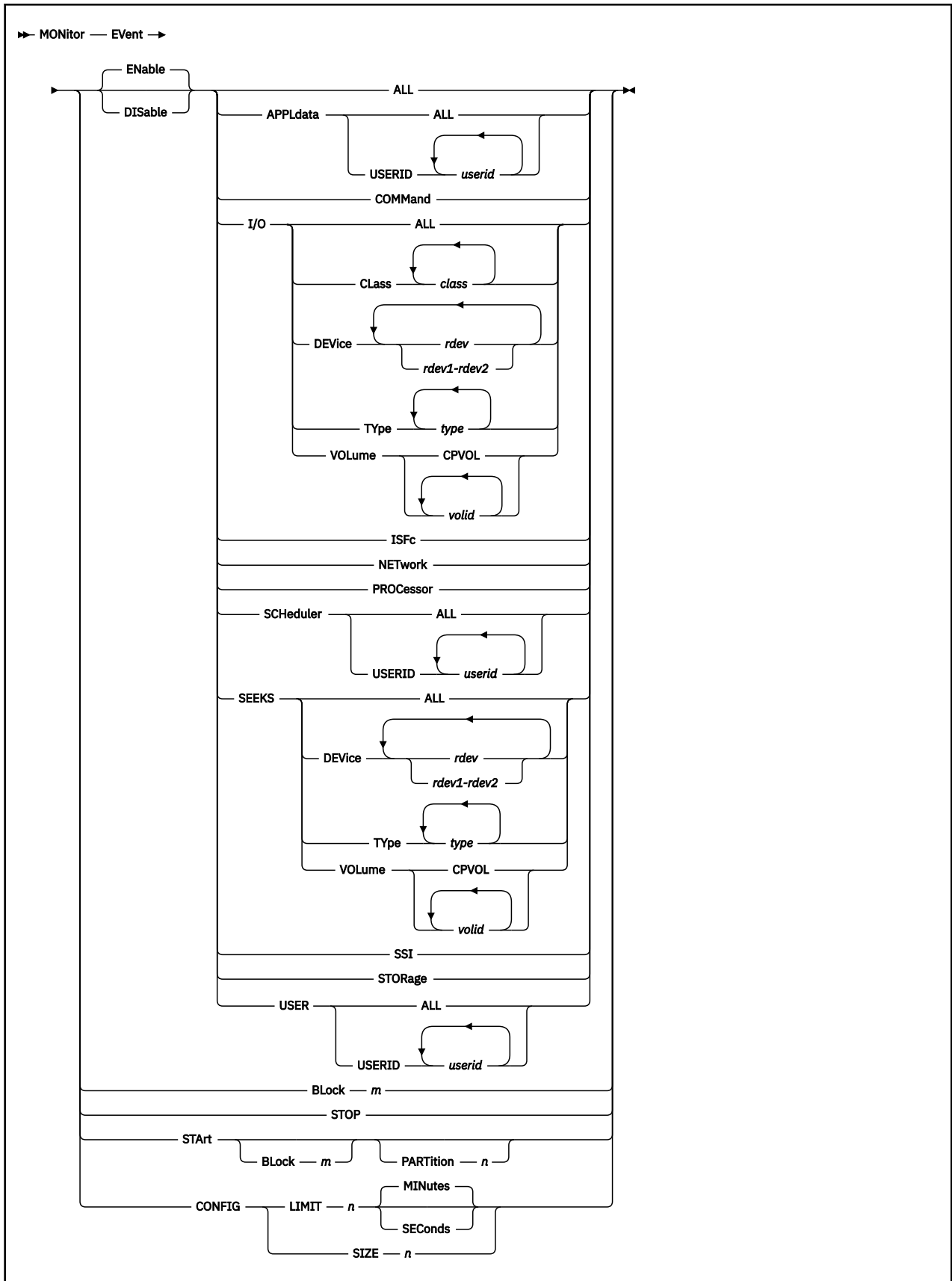
- a. FB-512 represents all FBA-type DASD devices.
  - b. GPRT represents all graphic display printers; GTERM represents all display terminals. For a list of graphic devices supported, see *z/VM: General Information*.
6. If while running monitor you experience missing or incomplete monitor records, the size of the sample configuration record area may not be large enough to record all the system configuration data, especially if there are a large number of devices available on your system. To increase the size of the sample configuration record area, see the MONITOR SAMPLE CONFIG command.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP007E Invalid userid - *userid*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP023E VOLID missing or invalid
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP053E *userid* not in CP directory
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1002E Volume identifier *valid* does not exist.
- HCP1550E *value* is an invalid device class.
- HCP6224I {*Sample*|*Event*} recording is pending because there are no users connected to \*MONITOR for this type of data.
- HCP6227E Monitor {*event*|*sample*} collection is not active.
- HCP6228E Monitor sample collection cannot start because DCSS *dcssname* is completely partitioned for event data.
- HCP6229E Monitor {*event*|*sample*} collection is already active.
- HCP6230E Monitor event collection cannot start because the block size is greater than half the event data area.

- HCP6231E Monitor event collection cannot start because the partition size is larger than DCSS *dcssname* size.
- HCP6233E Monitor event collection cannot start because the partition size equals the size of DCSS *dcssname*.
- HCP6234E *value* is an invalid monitor block value.
- HCP6235E *value* is an invalid monitor partition value.
- HCP6236E *value* is an invalid monitor interval value.
- HCP6237I Event monitoring will not begin until the event pages in the DCSS are available.
- HCP6238E *value* is an invalid monitor rate value.
- HCP6240I Device *device* is already enabled for monitoring.
- HCP6240I Class *class* is already enabled for monitoring.
- HCP6241I The number of devices successfully enabled for monitoring is *nnnn*. The number of devices enabled but offline is *nnnn*.
- HCP6241I Class *class* is enabled for monitoring.
- HCP6242I The class or type of device *device* is invalid for SEEKS monitoring
- HCP6246E Monitor has been terminated due to a soft abend.
- HCP6249I Device *rdev* is enabled for monitoring, but is offline
- HCP6250I A monitor sample interval has been skipped because the IUCV message limit of number to \*MONITOR has been reached.
- HCP6255I number monitor sample interval(s) have been skipped because data collection could not complete before the next sample interval.
- HCP6262E *value* is an invalid time limit for monitor {event|sample} CONFIG.
- HCP6274I The {sample|event} data messages and corresponding records have been purged.
- HCP6551I User *userid* will not be monitored for the APPLDATA domain because it is not authorized
- HCP6552E The size of the {event|sample} configuration area cannot be changed because {event|sample} monitoring has already been started.
- HCP6553E *value* is an invalid size for monitor {event|sample} CONFIG.
- HCP6554E The MONITOR EVENT START failed because the event partition size of *n* pages is not large enough. At least *n* pages are currently required.
- HCP6555E The MONITOR EVENT START failed because even with the largest possible DCSS, the default partition size will not be large enough. At least *n* pages are currently needed.
- HCP6556E {Sample|Event} monitoring cannot start because there are not enough pages available in the DCSS *dcssname*. *n* pages are available.
- HCP6557E {Sample|Event} monitoring cannot start because even the largest possible DCSS won't have enough pages. At most, *n* pages will be available.
- HCP6558E The MONITOR EVENT START failed because the default partition size cannot be allocated. Only *n* pages are available.
- HCP6559E The MONITOR EVENT START failed because even with the largest possible DCSS, the default partition size could not be allocated. At most, *n* pages are available.

# MONITOR EVENT



## Authorization

Privilege Class: A, E

## Purpose

Use MONITOR EVENT to establish or modify a profile for event data collection. You can also use this command to begin or end event monitoring. An event data profile lists:

- Data domains and domain elements to be monitored
- The number of pages reserved for event configuration records
- The block and partition values the shared segments.

**Note:** MONITOR EVENT START/STOP does *not* affect the profile and is *not* a subset of MONITOR EVENT ENABLE/DISABLE.

## Operands

### ENable

enables the specified domains and their elements for event data collection. Event data is collected only from those domains that have been enabled. ENABLE is the default.

**Note:** The MONITOR domain is always enabled for event monitoring.

### DISable

disables the specified domains or their elements from event data collection.

**Note:** The MONITOR domain cannot be disabled from monitoring.

### ALL

specifies that all domains and all their elements will be enabled or disabled for event data collection, including:

- APPLDATA (all authorized users)
- I/O (all devices)
- ISFC
- NETWORK
- PROCESSOR
- SCHEDULER (all users)
- SEEKS (all DASD devices)
- SSI
- STORAGE
- USER (all users)
- COMMAND

**Note:** The MONITOR domain is enabled for event data collection at all times.

### APPLdata

specifies that the APPLDATA domain will be enabled or disabled for event data collection. APPLDATA domain is where CP collects data generated by an application program in a virtual machine. The virtual machine issues DIAGNOSE X'DC' START to declare a buffer from which CP can collect the information. For information on DIAGNOSE Code X'DC' – Control Application Monitor Record Collection in *z/VM: CP Programming Services*.

### APPLDATA ALL

specifies that all authorized users will be enabled or disabled for event monitoring in the APPLDATA domain. To be authorized for monitoring in the APPLDATA domain, the user's directory entry must contain an OPTION APPLMON statement.

**APPLdata USERID *userid***

specifies a list of users to be enabled or disabled for event monitoring in the APPLDATA domain. The *userid* is a list of one or more user IDs, separated by blanks, from whom data is collected. Each user specified in this list must have the OPTION APPLMON statement in their directory entry. Users that do not have this statement in their directory entries will not be enabled for this domain.

**COMMANd**

specifies that the COMMAND synthetic domain will be enabled or disabled for event data collection. The COMMAND synthetic domain comprises the events in other domains that are produced by commands such as VARY ON and OFF, SET SHARE, and LOGOFF. This domain can be enabled to allow these records to be collected without also having to collect records that are generated at much higher frequency and produce substantial amounts of data.

**I/O**

specifies that the I/O domain will be enabled or disabled for event data collection. An I/O domain is comprised of all real I/O devices. Events in this domain include such activities as VARY ON and OFF, and attachment and detachment of devices. You may enable or disable all or selected I/O devices for the I/O domain. You can restrict them by device number, class type, or volume.

**I/O ALL**

specifies that all real devices will be enabled or disabled for event monitoring in the I/O domain.

**I/O CLass *class***

specifies a list of device classes that will be enabled or disabled for event monitoring in the I/O domain. *class* is a list of one or more device classes, separated by blanks. All real devices of the specified class or classes are monitored. For a list of classes that can be specified in this command, see [Table 31 on page 707](#).

**I/O DEVIce *rdev*****I/O DEVIce *rdev1-rdev2***

specifies the real devices to be enabled or disabled for event data collection in the I/O domain. You may specify the device number of a single device; you can also specify a list, a range of devices, or a combination of both. If you specify a range, all valid devices in the range, inclusively, are enabled or disabled. For additional information, see the specific usage notes for I/O.

**I/O TYpe *type***

specifies a list of device types to be enabled or disabled for event monitoring in the I/O domain. *type* is a list of one or more device types (such as 3380), separated by blanks. For valid device types, see [Table 31 on page 707](#). All real devices of the specified type or types are monitored.

**I/O VOLume *valid*****I/O VOLume CPVOL**

specifies a list of volumes that are to be enabled or disabled for event monitoring in the I/O domain. *valid* is a list of one or more volume identifiers, separated by blanks. If CPVOL is specified, all CP-owned volumes will be enabled or disabled for monitoring. A CP-owned volume is any volume specified on the CP\_OWNED statement in the system configuration file. For devices identified by DEVNO, you cannot use the volume option of Monitor. The volume option can only be used for devices identified by VOLSER.

**IFSc**

specifies that the ISFC domain is to be enabled or disabled for event data collection. Data such as ISFC Transport end point status change or ISFC Logical Link Definition changes are recorded.

**NETwork**

specifies that the NETWORK domain is to be enabled or disabled for event data collection. Data such as a virtual guest adapter initialization or termination is recorded.

**PROCeSSor**

specifies that the PROCESSOR domain will be enabled or disabled for event data collection. Events such as VARY ON or OFF of a processor are monitored.

**SCHeduler**

specifies that the SCHEDULER domain will be enabled or disabled for event data collection. Event data include such statistics as scheduler queue manipulation, flow of work through the system, and resource allocation strategies of the scheduler and the dispatcher.



**SCheduler ALL**

specifies that all users will be enabled or disabled for event data collection in the SCHEDULER domain.

**SCheduler USERID *userid***

specifies a list of users that will be enabled or disabled for event data collection in the SCHEDULER domain. *userid* is a list of 1 or more user IDs, separated by blanks.

**SEEKS**

specifies that the SEEKS domain will be enabled or disabled for event data collection. A SEEKS domain is comprised of all architected DASD devices. Event data, which include DASD positioning, are collected on every I/O request on such enabled DASD devices. You may enable or disable all or specified devices in the SEEKS domain. You can restrict them by their device numbers, types, or volumes. Event data are collected on every I/O request on such enabled DASDs.

**SEEKS ALL**

specifies that all real DASDs will be enabled or disabled for event data collection in the SEEKS domain.

**SEEKS DEvice *rdev*****SEEKS DEvice *rdev1-rdev2***

specifies the DASDs to be enabled or disabled for event data collection in the SEEKS domain. You may specify the device number of a single device; you can also specify a list, a range of devices, or a combination of both. If you specify a range, all valid DASD devices in the range, inclusively, are enabled or disabled. See the specific SEEKS usage notes for additional information.

**SEEKS Tpe *type***

specifies a list of DASD types that will be enabled or disabled for event data collection in the SEEKS domain. *type* is a list of one or more device types (such as a 3380), separated by blanks. All DASDs of the specified type or types are monitored. For valid device types, see [Table 31 on page 707](#).

**SEEKS VOLume *valid*****SEEKS VOLume CPVOL**

specifies a list of volumes that will be enabled or disabled for event data collection in the SEEKS domain. *valid* is a list of one or more volume identifiers, separated by blanks. If CPVOL is specified, all CP-owned volumes are enabled or disabled. A CP-owned volume is any volume specified on the CP\_OWNED statement in the system configuration file. For devices identified by DEVNO, you cannot use the volume option of Monitor. The volume option can only be used for devices identified by VOLSER.

**SSI**

specifies that the SSI domain will be enabled or disabled for event data collection. SSI domain events include z/VM Single System Image (SSI) cluster state changes and slot definition data. The SSI operand cannot be specified if the z/VM system is not part of the z/VM SSI cluster.

**STORAge**

specifies that the STORAGE domain will be enabled or disabled for event data collection. Events such as defining and purging of a named saved system or a saved segment, or attaching, or detaching a CP volume are monitored.

**USER**

specifies that the USER domain will be enabled or disabled for event data collection. A USER domain is comprised of all users. All or selected users may be specified for monitoring. USER domain events include such activities as logon and logoff, defining and detaching a virtual CPU, and an end of transaction.

**USER ALL**

specifies that all users will be enabled or disabled for event data collection in the USER domain.

**USER USERID *userid***

specifies a list of users to be enabled or disabled in the USER domain. *userid* is a list of one or more user IDs, separated by blanks.

**Event BLock *m***

specifies the maximum number of 4 KB pages, *m*, of the shared segment (saved segment), that are accumulated before the virtual machine is notified that the event data is available for data collection. You can modify *m* using the MONITOR EVENT BLOCK command while event monitoring is active.

### Event STOP

deactivates the event-driven data collection. The monitor event profile is not affected by this command, except for PARTITION, which is set to zero.

**Note:** Do not use this command to stop the MONWRITE utility. (See [“MONWRITE” on page 2212](#) for more information.)

### Event START

starts event monitoring for the domains and their elements listed in the event profile set by the preceding MONITOR EVENT commands.

If no virtual machine is connected for event data (or no virtual machine connected for event data has an active path to \*MONITOR) when the command is issued, event recording is held pending.

When recording is activated (that is, when there is a user connected with an active path for event data), event configuration records from the monitor domain are the first event records generated.

### BLOCK *m*

specifies the maximum number of 4 KB frames of the shared segment, that are accumulated before the virtual machine is notified that the event data is available for data collection.

The BLOCK value may be modified using the MONITOR EVENT BLOCK command while event monitoring is active.

### PARTITION *n*

specifies the number of 4 KB pages in the shared segment to be reserved for event records. The rest of the shared segment is used for sample data.

### EVENT CONFIG LIMIT *n* MINutes

### EVENT CONFIG LIMIT *n* SECOnds

specifies the time limit a user can refer to the event configuration area before the data is refreshed or zeroed.

### EVENT CONFIG SIZE *n*

specifies the size in 4 KB pages to be allocated in the saved segment for EVENT configuration records.

## Usage Notes

(General)

1. Both event and sample monitoring can be active at the same time. The MONITOR EVENT START command starts only event monitoring. To start both event and sample monitoring, see [“MONITOR START/STOP” on page 727](#). Similarly, MONITOR EVENT STOP, stops only event monitoring while MONITOR STOP stops both event and sample monitoring.
2. The following must be done to activate event recording:
  - A saved segment must be defined and saved for the monitor
  - A virtual machine must load the monitor saved segment and connect to \*MONITOR for event data
  - A MONITOR START or MONITOR EVENT START command must be issued.

If the START command is issued when no virtual machine is connected with an active path to \*MONITOR for event data, event recording is held pending until there is a connection with an active path.

3. The event configuration message may be delayed for a user who connects to \*MONITOR to collect event data and, in turn, the event data notification messages are also delayed for the user.
  - If another user has already connected to \*MONITOR to collect event data and has not replied to the event configuration message, the monitor regenerates the configuration records after the configuration area has been freed, and sends an event configuration message to the user who has just connected to \*MONITOR
  - If a user's path is quiesced (or at its IUCV message limit) when the event configuration notification is sent, the user receives notification for a refreshed set of configuration records when the user resumes the path to \*MONITOR (or responds to one or more previous messages).

4. The event profile can be changed after monitoring has started. The only exceptions are that the partition size and the event config size of the saved segment cannot be changed unless event monitoring is stopped. New event domains are activated upon completion of the event command.
5. Event monitoring can result in large volumes of data generated and an increased CP overhead, particularly in the SEEKS and SCHEDULER domains. For this reason, these two domains should be enabled only when needed and should be limited to specific devices or a specific set of devices.

#### Specific Usage Notes for I/O and SEEKS:

1. Any overlapping of I/O devices can exist in a series of MONITOR EVENT commands on the I/O domain and the SEEKS domain. For example, a command:

```
monitor event enable i/o type 3380
```

followed by:

```
monitor event enable i/o class dasd
```

would result in all DASD's enabled for monitoring. If this is then followed by:

```
monitor event disable i/o type 3380
```

monitoring ceases for all 3380 devices but continues for all other types of DASD devices.

2. A message is issued for any device that had been specified but does not exist. For the SEEKS domain, a message is issued for any specified device that is found not to be a DASD.
3. Devices that are enabled but are found to be offline are monitored after they are varied online. A summary message is issued indicating how many devices are enabled for monitoring, and how many are offline.
4. The VOLUME operand can be used only for volumes attached to the system, because CP associates a volume label to a device number only at the time it is attached. If you want to enable a DASD device that is not attached to the system, use the DEVICE option.
5. The volume to be monitored is maintained by its device address. Enabling or disabling a volume for monitoring remains valid as long as its device address does not change. For example, if a volume is detached and a volume with the same label at a different address is attached, the volume must be enabled again to be monitored and its previous device address disabled.

#### Specific Usage Notes for BLOCK:

1. If BLOCK is omitted, the default is four 4 KB pages.
2. The minimum value for BLOCK is four 4 KB pages.
3. The maximum value for BLOCK is half the number of pages to be used for event data (partition-event config size).
4. The value of BLOCK is preserved when monitoring is stopped.

#### Specific Usage Notes for CONFIG SIZE:

1. The default size reserved for event configuration records when event monitoring is started is 68 pages.
2. The MONITOR EVENT CONFIG SIZE can be issued only when event monitoring is not active. To change the size after event monitoring has been started, you must first stop event monitoring.
3. The minimum value for CONFIG SIZE is one 4 KB page.
4. The maximum value for CONFIG SIZE is equal to the maximum number of pages that can be specified as a type SC range with the DEFSEG command, minus 8 pages for event data records.

**Note:** The number of pages reserved must fit within the saved segment to be used by monitor. This check is made when monitoring is started (if someone is connected to \*MONITOR and the saved segment is known) or when the first user connects after monitoring has been started.

5. The value of CONFIG SIZE is preserved when monitoring is stopped.

#### Specific Usage Notes for CONFIG LIMIT:

## MONITOR EVENT

1. The CONFIG time limit is the amount of time a user can keep the configuration area in use.
2. The CONFIG time limit default is 1 minute.
3. The minimum value for CONFIG time limit is 1 second.
4. The maximum value for CONFIG time limit is 120 minutes.
5. The CONFIG time limit value must be specified as a whole number.
6. When the CONFIG time limit expires, CP:
  - In the EXCLUSIVE environment—stops monitoring and issues message HCP6239I
  - In the SHARED environment—purges the configuration message, releases the corresponding saved-segment pages, and issues message HCP6253I. Data recording continues.
7. Configuration records are created and the CONFIG time limit enforced whenever:
  - Recording has begun. (A MONITOR START command is issued and at least one user is connected, with an active path, for the type of monitoring being started.)
  - Recording is underway and:
    - A user connects (with an active path) for that type of monitor data
    - The path of a user connected for that type of data becomes active
    - The user, whose path to \*MONITOR was quiesced when \*MONITOR tried to send a configuration notification message, issues IUCV RESUME
    - The user, whose path to \*MONITOR was at its IUCV message limit when \*MONITOR tried to send a configuration notification message, issues IUCV REPLY or REJECT to one or more previous messages from \*MONITOR.
8. CONFIG time limit is preserved when monitoring is stopped.

### Specific Usage Notes for PARTITION:

1. The default value of PARTITION is half of the size of the saved segment.
2. The minimum PARTITION size is 8 pages plus the number of pages reserved for event configuration records. The absolute minimum partition size is 9 pages.

#### Note:

- a. To determine the number of pages currently designated for event configuration records, enter the QUERY MONITOR EVENT command.
  - b. To reserve a number of pages for the event configuration area, enter the MONITOR EVENT CONFIG SIZE command. If the MONITOR EVENT CONFIG SIZE command has not been issued when the MONITOR EVENT START command is issued, 68 pages are reserved for event configuration recording, and the PARTITION size must be at least 76 pages.
3. The maximum PARTITION size depends on whether sample monitoring has already been started, and on whether any user is connected to \*MONITOR when the command is issued to start event monitoring.

If sample monitoring has *not* started, and:

- If one or more users are connected, the maximum PARTITION value that can be entered is the number of pages in the first type SC range of the saved segment that is currently being used by monitor.
- If there are no users connected to \*MONITOR, the maximum value for PARTITION is equal to the maximum number of pages that can be specified as a type SC range with the DEFSEG command.

If sample monitoring *has* been started, the maximum number of pages that can be specified by PARTITION is reduced by one more than the number of pages reserved for sample configuration records. (QUERY MONITOR EVENT or QUERY MONITOR SAMPLE displays the number of pages currently reserved for the event and the sample configuration areas.)

4. When monitoring is stopped, the value of PARTITION reverts to the default value.

## Responses

The command MONITOR EVENT START can be issued before or after a user connects to \*MONITOR to collect event data.

Response 1:

```
MONITOR EVENT STARTED -- RECORDING IS ACTIVATED
```

- or -

```
MONITOR EVENT STARTED BY [userid] -- RECORDING IS ACTIVATED
```

As a result of the MONITOR EVENT START command, event monitoring has successfully started. There is at least one user with an active path connected to \*MONITOR for event data collection.

This response is sent to the user that issued the command. If this user is not the system operator, the response containing [userid] is also sent to the system operator.

Response 2:

```
MONITOR EVENT STARTED -- RECORDING IS PENDING
```

- or -

```
MONITOR EVENT STARTED BY [userid] -- RECORDING IS PENDING
```

MONITOR EVENT START has been issued. Because there is no user with an active path connected to \*MONITOR for event data collection, event monitoring is held pending. It shall remain pending until the first user is connected to \*MONITOR for event data collection.

This response is sent to the user that issued the command. If this user is not the system operator, the response containing [userid] is also sent to the system operator.

Response 3:

```
MONITOR EVENT STOPPED
```

- or -

```
MONITOR EVENT STOPPED BY [userid]
```

MONITOR EVENT STOP has been issued and event monitoring has been stopped.

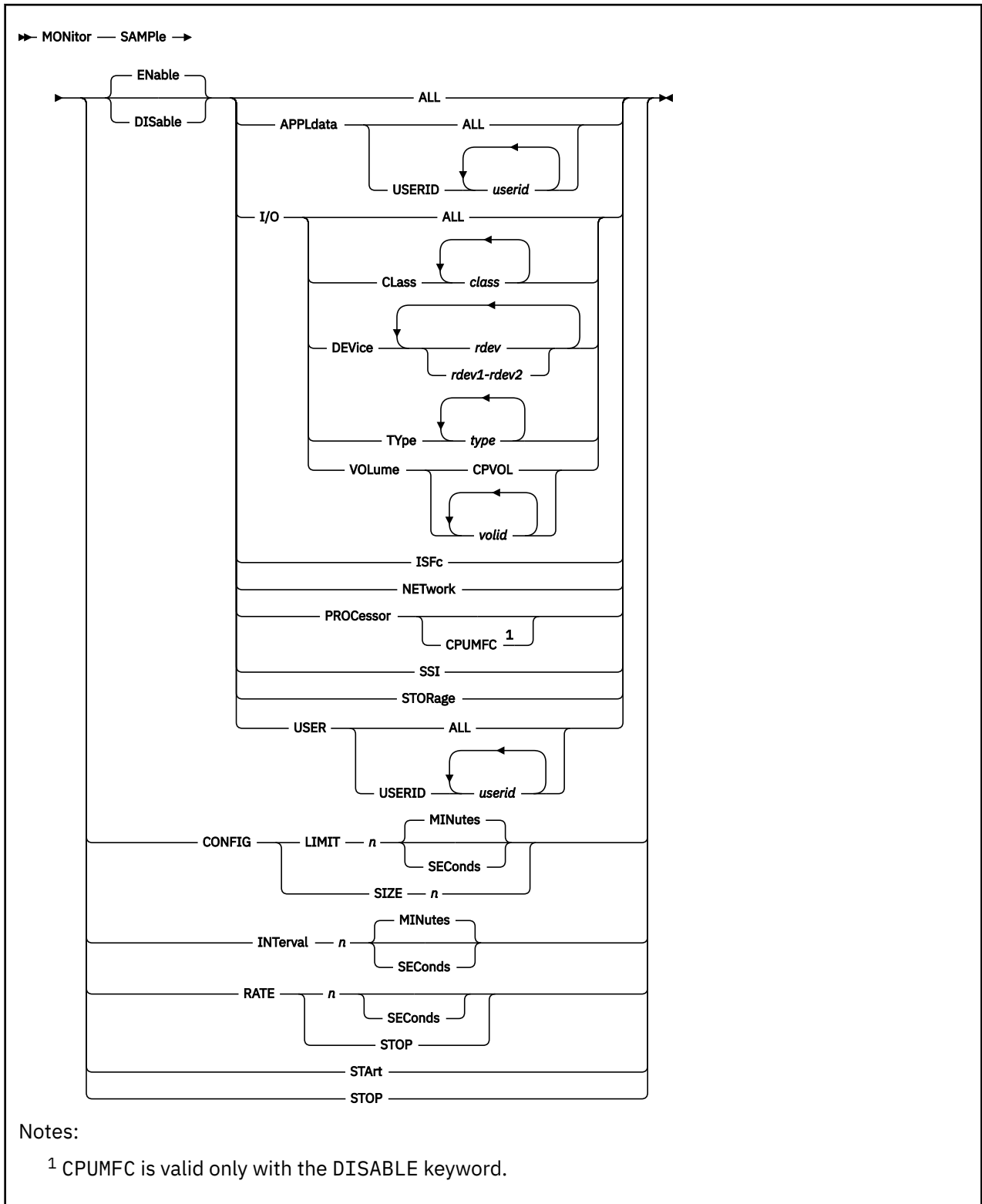
This response is sent to the user that issued the command. If this user is not the system operator, the response containing [userid] is also sent to the system operator.

Response 4:

```
Command complete
```

The MONITOR command has completed successfully.

# MONITOR SAMPLE



## Authorization

Privilege Class: A, E

## Purpose

Use MONITOR SAMPLE to establish or modify a profile for collecting both single sample and high-frequency sample data. This command can also be used to begin or end sample monitoring.

Single sample data are a set of counters and data that characterize a system operation. High-frequency sample data are a set of counters and data that characterize the system at the moment the data is sampled.

The sample profile lists the following:

- Data domains and domain elements to be monitored
- Rate at which high-frequency sample data is collected
- Time interval at which sample data is reported
- Number of pages to be reserved for sample configuration records.

## Operands

### ENable

enables the specified domains and their elements for sample data collection. Sample data is collected only from those domains that have been enabled. ENABLE is the default.

**Note:** The MONITOR domain and the SYSTEM domain are always enabled for sample data monitoring.

### DISable

disables the specified domains or their elements from monitoring.

**Note:** The MONITOR domain and the SYSTEM domain cannot be disabled from sample data collection.

### ALL

specifies that all domains and all their elements are to be enabled or disabled for sample data collection. This includes the following domains:

- APPLDATA (all authorized users)
- I/O (all devices)
- ISFC
- NETWORK
- PROCESSOR
- SSI
- STORAGE
- USER (all users)

**Note:** SYSTEM and MONITOR domains are enabled for sample monitoring at all times.

### APPLdata

specifies that the APPLDATA domain is to be enabled or disabled for sample data collection. APPLDATA domain is where CP collects data generated by an application program in a virtual machine. The virtual machine issues DIAGNOSE X'DC' START to declare a buffer from which CP can collect the information. For information on [DIAGNOSE Code X'DC' – Control Application Monitor Record Collection](#) in *z/VM: CP Programming Services*.

### APPLdata ALL

specifies that all authorized users are to be enabled or disabled for sample monitoring in the APPLDATA domain. To be authorized for monitoring in the APPLDATA domain, the user's directory entry must contain an OPTION APPLMON statement.

### APPLdata USERID *userid*

specifies a list of users to be enabled or disabled for sample monitoring in the APPLDATA domain. The *userid* is a list of one or more user identifiers, separated by blanks, from whom data is collected.

Each user specified in this list must have the OPTION APPLMON statement in their directory entry. Users who do not have this statement in their directory entries are not enabled for this domain.

**I/O**

specifies that the I/O domain is to be enabled or disabled for sample data collection. An I/O domain is comprised of all real I/O devices. You may enable or disable all or selected I/O devices for the I/O domain. You can restrict them by device numbers, classes, types, or volumes.

**I/O ALL**

specifies that all real devices are to be enabled or disabled for sample data collection in the I/O domain.

**I/O CClass *class***

specifies a list of device classes that are to be enabled or disabled for sample data collection in the I/O domain. *class* is a list of one or more device classes, separated by blanks. All real devices of the specified class or classes are monitored.

For a list of classes that can be specified in this command, and for a list of device types that each class supports, see [Table 31 on page 707](#).

**I/O DEvIce *rdev*****I/O DEvIce *rdev1-rdev2***

specifies the real devices to be enabled or disabled for sample data collection in the I/O domain. You may specify the device number of a single device; you can also specify a list, a range of devices, or a combination of both. If you specify a range, all valid devices in the range, inclusively, are enabled or disabled. For additional information, see [Specific Usage Notes for I/O](#).

**I/O TYpe *type***

specifies a list of device types that are to be enabled or disabled for sample data collection. *type* is a list of one or more device types (such as the 3380), separated by blanks. For valid device types, see [Table 31 on page 707](#). All real devices of the specified type or types are monitored.

**I/O VOLume *valid*****I/O VOLume CPVOL**

specifies a list of volumes that are to be enabled or disabled for sample data collection. *valid* is a list of one or more volume identifiers, separated by blanks. If CPVOL is specified, all CP-owned volumes are to be enabled or disabled for monitoring. A CP-owned volume is any volume specified on the CP\_OWNED statement in the system configuration file. For devices identified by DEVNO, you cannot use the VOLUME option of MONITOR. The VOLUME option can only be used for devices identified by VOLSER.

**IFSc**

specifies that the ISFC domain is to be enabled or disabled for sample data collection. Data such as ISFC Transport end point status change or ISFC Logical Link Definition changes are recorded.

**NETwork**

specifies that the NETWORK domain is to be enabled or disabled for sample data collection. Data such as the number of packets sent by and delivered to a virtual guest adapter are recorded.

**PROcessor**

specifies that the PROCESSOR domain, including the CPU-Measurement Counter Facility, is to be enabled or disabled for sample data collection. Data related to CPU usage such as system locks, simulation, and interprocessor signaling are collected.

**PROcessor CPUMFC**

specifies that collection of CPU-Measurement Counter Facility counters is to be disabled for counter sets requiring activation by CP. Sample data collection of the rest of the PROCESSOR domain is unaffected.

Collection of CPU-Measurement Counter sets for the Multithreading (MT) Facility is unaffected. Those counter sets are collected when multithreading is enabled and collection of monitor sample records in the PROCESSOR domain is enabled.

**Note:** The CPUMFC keyword is valid only with the DISABLE keyword.



**SSI**

specifies that the SSI domain will be enabled or disabled for sample data collection. Data collected for this domain includes z/VM Single System Image (SSI) cluster state change synchronization activities and mode occupancy. The SSI operand cannot be specified if the z/VM system is not part of the z/VM SSI cluster.

**STORAge**

specifies that the STORAGE domain is to be enabled or disabled for sample data collection. Status and utilization data of real, virtual, shared, and auxiliary storage are collected.

**USER**

specifies that the USER domain is to be enabled or disabled for sample data collection. A USER domain is comprised of all users. Data collected for this domain consists of user resource utilization and status data for all or selected users.

**USER ALL**

specifies that all users are to be enabled or disabled for sample data collection in the USER domain.

**USER USERID *userid***

specifies a list of users that are to be enabled or disabled for sample data collection. *userid* is a list of one or more user identifiers, separated by blanks.

**SAMPlE INTerval *n* MINutes****SAMPlE INTerval *n* SECOnds**

specifies the time interval at which the sample data is to be reported. MINUTES is the default. For additional information, see [Specific Usage Notes for INTERVAL](#).

**SAMPlE RATE *n* SECONDS**

specifies the frequency at which the high-frequency sample data is to be sampled. The operand *n* is the length of time between high-frequency sampling. High-frequency sample data is a set of counters and data that represent the state of the system at the moment they are sampled.

High-frequency sampling is automatically activated whenever the MONITOR SAMPLE START command is issued, unless the RATE value has been previously set to STOP. For additional information, see [Specific Usage Notes for RATE](#).

**SAMPlE RATE STOP**

immediately terminates the sampling of high-frequency data. Data for the current interval is lost.

**SAMPlE CONFIG SIZE *n***

specifies the size in 4 KB pages to be allocated in the saved segment for SAMPLE configuration records.

**SAMPlE CONFIG LIMIT *n* MINutes****SAMPlE CONFIG LIMIT *n* SECOnds**

specifies the time limit that the configuration records remain in the CONFIG area of the saved segment before they are refreshed or zeroed.

**SAMPlE STArt**

starts sample monitoring for the domains and their elements specified in preceding MONITOR SAMPLE commands.

If no application is connected (with an active path) for sample data when the command is issued, sample recording is held pending. However, high-frequency sampling, if enabled, is started.

When recording is activated (that is, when there is a user connected with an active path for sample data), sample configuration records from the monitor domain are generated. Sample configuration records include information concerning the processor, storage, paging, real I/O configurations, and initial sample configuration settings.

**SAMPlE STOP**

deactivates sample monitoring. Collection of single sample data and high frequency data are stopped. All data collected for the current interval are purged. The monitor sample profile is not affected by this command.

**Note:** Do not use this command to stop the MONWRITE utility. (See [“MONWRITE” on page 2212](#) for more information.)

## Usage Notes

(General)

1. The following must be done to activate sample recording:
  - A saved segment must be defined and saved for the monitor
  - A virtual machine must load the monitor saved segment and connect to \*MONITOR for sample data
  - A MONITOR START or MONITOR SAMPLE START command must be issued.

If the START command is issued when no virtual machine is connected with an active path to \*MONITOR for sample data, sample recording is held pending until there is a connection with an active path.
2. The intervals for sample data collection start at the time the sample start command is issued. Monitor does not generate sample records unless there is at least one eligible monitor user for sample data. The following are two examples for the condition mentioned above:
  - Example 1
    - a. MONITOR SAMPLE START is issued at 10:00 A.M. with 10-minute intervals.
    - b. A user connects to \*MONITOR at 10:05 A.M. to collect sample data. The sample configuration records are generated at the connect time (10:05 A.M.), and the first data records are generated at 10:10 A.M.
  - Example 2
    - a. A user connects to \*MONITOR at 10:00 A.M. to collect sample data.
    - b. MONITOR SAMPLE START is issued at 10:05 A.M. with 10-minute intervals. The first sample data records are generated immediately after the sample configuration records are generated (at 10:05 A.M.). The next sample interval is at 10:15 A.M.
3. It is possible for a user who connects to \*MONITOR for sample data collection to receive two record notification messages—one for sample configuration followed by one for the sample data records. See [Example 2](#) above.
4. The sample configuration message might be delayed for a user who connects to \*MONITOR to collect sample data, and in turn the sample data notifications are also delayed for the user.
  - If another user has already connected to \*MONITOR to collect sample data and has not replied to the sample configuration message, the monitor regenerates the configuration records after the configuration area is free and sends a sample configuration message to the user who just connected to \*MONITOR.

**Note:** If the MONITOR SAMPLE START command is issued before any user is connected to \*MONITOR, the first user connected to \*MONITOR holds the configuration area. When that user issues an IUCV REPLY or the configuration time limit expires, the other users who are connected receive data notification for a new set of configuration records.

  - If a user's path to \*MONITOR is quiesced (or at its IUCV message limit) when the sample configuration notification is sent, the user receives notification for a refreshed set of configuration records when the user resumes the path (or responds to one or more previous messages).
5. Both event and sample monitoring can be active at the same time. The MONITOR SAMPLE START command only starts sample monitoring. To start both event and sample monitoring, use the MONITOR START command described in “MONITOR START/STOP” on page 727. Similarly, MONITOR SAMPLE STOP stops only sample monitoring while MONITOR STOP stops both event and sample monitoring.

Specific Usage Notes for I/O:

1. Any overlapping of I/O devices can exist on a series of MONITOR SAMPLE commands on the I/O domain. For example, if the command:

```
monitor sample enable i/o type 3380
```

is followed by:

```
monitor sample enable i/o class dasd
```

all DASD are enabled for monitoring. If the command is then followed by:

```
monitor sample disable i/o type 3380
```

monitoring ceases for all 3380 devices, but continues for all other types of DASD.

2. A message is issued for any specified device that does not exist.
3. Devices that are enabled but are found to be offline are monitored when varied online. A summary message is issued indicating how many devices are enabled for monitoring, and how many are offline.
4. The VOLUME operand can be used only for volumes attached to the system, because CP associates a volume label to a device number only at the time it is attached. If it is desired to enable a DASD not attached to the system, it can be done by using the DEVICE option.
5. The volume to be monitored is maintained by its device address. Enabling or disabling a volume for monitoring remains valid as long as its device address does not change. For example, if a volume is detached and a volume with the same label at a different address is attached, the volume must be enabled again to be monitored, and its previous device address disabled.
6. A cache data record will not be created for a secondary device of a duplex pair.

Specific Usage Notes for PROCESSOR:

1. The MONITOR SAMPLE ENABLE PROCESSOR command can be used to turn on collection of CPU-Measurement Counter Facility counters only if the machine supports the CPU-Measurement Counter Facility, the z/VM software recognizes and supports the counter first version number (CFVN) and counter second version number (CSVN) values reported by the hardware, and the partition is authorized for the basic counter set. Authorization is set either in the partition's activation profile or through the Change LPAR Security task on the Support Element. If one or more of these prerequisites is not met, z/VM does not activate the collection of CPU-Measurement Counter Facility counters. If all of these prerequisites are met, z/VM collects counters in the basic counter set, along with counters in the problem-state, crypto-activity, and extended counter sets, provided those counters are authorized. See the CPUMFC operand for an explanation of how the collection of these counter sets can be controlled.

The MT-Diagnostic counter set is not controlled by the CPUMFC operand. This counter set is authorized, enabled, and activated only if the Multithreading Facility is enabled. Therefore it is always collected if the Multithreading Facility is enabled and the PROCESSOR domain is enabled for sample data monitoring.

2. Message HCP6263I is displayed when CPU-Measurement Counter Facility counters are available on the machine but the basic counter set is not authorized for the partition.
3. The CPU-Measurement Counter Facility counters are part of the PROCESSOR sample domain. However, you might want to run with the PROCESSOR sample domain enabled but with the CPU-Measurement Counter Facility counters turned off. With respect to the PROCESSOR domain, the MONITOR SAMPLE command has the following behaviors:
  - a. MONITOR SAMPLE ENABLE PROCESSOR turns on the entire PROCESSOR sample domain including the CPU-Measurement Counter Facility counters.
  - b. MONITOR SAMPLE DISABLE PROCESSOR turns off the entire PROCESSOR sample domain, including the CPU-Measurement Counter Facility counters.
  - c. MONITOR SAMPLE DISABLE PROCESSOR CPUMFC turns off collection of CPU-Measurement Counter Facility counters but leaves the rest of the PROCESSOR sample domain enabled.
4. For more information about CPU-Measurement Counter Facility counters and the contents and formats of the various counter sets, refer to *Load-Program-Parameter and CPU-Measurement Facilities* and *CPU-Measurement Facility Extended Counters Definition* accompanying the machine on which you are running.

Specific Usage Notes for INTERVAL:

## MONITOR SAMPLE

1. The valid range of values for INTERVAL is 6 to 3600 for SECONDS, and 1 to 60 for MINUTES. INTERVAL values must be specified in whole numbers.
2. The default value of INTERVAL is one minute (60 seconds).
3. If an interval is currently in effect, the new interval does not take effect until the next sample data record is produced.
4. The SAMPLE INTERVAL value must be greater than or equal to the SAMPLE RATE value.
5. It is possible that data may not be reported by the time the interval expires. This happens when it takes longer than an interval to collect the data. If this happens, the interval time should be adjusted to allow sufficient time to collect and report the data.

### Specific Usage Notes for RATE:

1. The valid range of value for RATE is .01 to 30 seconds. As many as two digits may be specified after the decimal point. Leading zeros are not required, and trailing zeros are not required after the decimal point.
2. The default value of RATE is two seconds.
3. If a rate is currently in effect, the new rate does not take effect until the current high-frequency rate has elapsed.
4. The SAMPLE RATE value specified must be less than or equal to the SAMPLE INTERVAL value.
5. The frequency of doing high-frequency sampling is determined by the RATE setting, however the creation of monitor records containing high-frequency data is controlled by MONITOR SAMPLE INTERVAL.

### Specific Usage Notes for CONFIG SIZE:

1. The default size reserved for sample configuration records when sample monitoring is started is 4096 pages.
2. The MONITOR SAMPLE CONFIG SIZE can be issued only when sample monitoring is not active. To change the size after sample monitoring has been started, you must first stop sample monitoring.
3. The minimum value for CONFIG SIZE is one 4 KB page.
4. The maximum value for CONFIG SIZE is equal to the maximum number of pages that can be specified as a type SC range with the DEFSEG command, minus 1 page for sample data records.

**Note:** The number of pages reserved must fit within the saved segment to be used by monitor. This check is made when monitoring is started (if someone is connected to \*MONITOR and the saved segment is known) or when the first user connects after monitoring has been started.

5. The value of CONFIG SIZE is preserved when monitoring is stopped.
6. If while running monitor you experience missing or incomplete monitor records, the size of the sample configuration record area may not be large enough to record all of the system configuration data, especially if there are a large number of devices available on your system. To increase the size of the sample configuration record area, see the MONITOR SAMPLE CONFIG command.

### Specific Usage Notes for CONFIG LIMIT:

1. The CONFIG time limit is the amount of time a user can keep the configuration area in use.
2. The default value for the CONFIG time limit is one minute.
3. The minimum value for CONFIG time limit is one second.
4. The maximum value for CONFIG time limit is 120 minutes.
5. The CONFIG time limit value must be specified as a whole number.
6. When the CONFIG time limit expires, CP:
  - In the EXCLUSIVE environment—stops monitoring and issues message HCP6239I.
  - In the SHARED environment—purges the configuration message, releases the corresponding saved segment pages, and issues message HCP6253I. Data recording continues.
7. Configuration records are created and the CONFIG time limit enforced whenever:

- Recording begins. (A MONITOR START command is issued and at least one user is connected, with an active path, for the type of monitoring being started.)
  - Recording is underway, and:
    - A user connects (with an active path) for that type of monitor data
    - The path of a user connected for that type of data becomes active
    - The user, whose path to \*MONITOR was quiesced when \*MONITOR tried to send a configuration notification message, issues IUCV RESUME
    - The user, whose path to \*MONITOR was at its IUCV message limit when \*MONITOR tried to send a configuration notification message, issues IUCV REPLY or REJECT to one or more previous messages from \*MONITOR.
8. CONFIG TIME LIMIT is preserved when monitoring is stopped.

## Responses

The command, MONITOR SAMPLE START, can be issued before or after a user connects to \*MONITOR to collect sample data.

Response 1:

```
MONITOR SAMPLE STARTED -- RECORDING IS ACTIVATED
```

- or -

```
MONITOR SAMPLE STARTED BY [userid] -- RECORDING IS ACTIVATED
```

As a result of the MONITOR SAMPLE START command, sample monitoring has successfully started. There is at least one user with an active path connected to \*MONITOR for sample data collection.

This response is sent to the user that issued the command. If this user is not the system operator, the response containing [userid] is also sent to the system operator.

Response 2:

```
MONITOR SAMPLE STARTED -- RECORDING IS PENDING
```

- or -

```
MONITOR SAMPLE STARTED BY [userid] -- RECORDING IS PENDING
```

MONITOR SAMPLE START has been issued. Because there is no user with an active path connected to \*MONITOR for sample data collection, sample monitoring is held pending. It shall remain pending until the first user is connected to \*MONITOR for sample data collection.

This response is sent to the user that issued the command. If this user is not the system operator, the response containing [userid] is also sent to the system operator.

Response 3:

```
MONITOR SAMPLE STOPPED
```

- or -

```
MONITOR SAMPLE STOPPED BY [userid]
```

MONITOR SAMPLE STOP has been issued and sample monitoring has been stopped.

This response is sent to the user that issued the command. If this user is not the system operator, the response containing [userid] is also sent to the system operator.

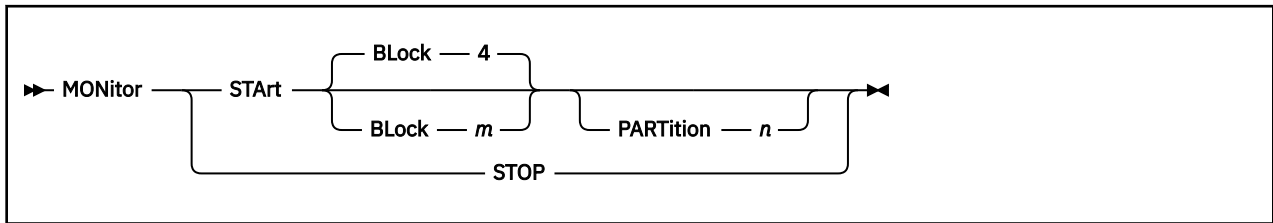
Response 4:

## MONITOR SAMPLE

```
Command complete
```

The MONITOR command has completed successfully.

## MONITOR START/STOP



### Authorization

Privilege Class: A, E

### Purpose

Use the MONITOR START command to activate event and sample monitoring based on the profiles established by the preceding MONITOR EVENT and MONITOR SAMPLE commands or by either one of those commands.

Use the MONITOR STOP command to deactivate all monitoring.

### Operands

#### STArT

activates event and sample monitoring based on the profiles established by the MONITOR EVENT and MONITOR SAMPLE commands. Event monitoring is started first.

#### STOP

deactivates the event and sample monitoring. The profiles for event and sample monitoring are not affected by this command, except for PARTITION which is set to zero. Event monitoring is stopped first.

**Note:** Do not use this command to stop the MONWRITE utility. (See “MONWRITE” on page 2212 for more information.)

#### BLoCk *m*

specifies the number of 4 KB frames, *m*, of the shared segment that are accumulated before the virtual machine is notified that the event data is available for data collection. The BLOCK value may be modified using the MONITOR EVENT BLOCK command while event monitoring is active.

#### PARTition *n*

specifies the number of 4 KB frames, *n*, in the shared segment that are reserved for event records. The rest of the shared segment is then used for sample data.

### Usage Notes

1. The MONITOR START command can be issued before or after a user connects to \*MONITOR. If the MONITOR START command is issued and there is no connection to \*MONITOR through an active path, monitor recording goes into a pending state. Monitor recording is not activated until there is a user connected with an active path (a path that is not quiesced at its IUCV message limit).

The following must be done before monitor recording gets activated:

- A saved segment must be defined and saved for the monitor
- A virtual machine must load the monitor saved segment and connect to the \*MONITOR CP system service
- A MONITOR START command must be issued.

## MONITOR START/STOP

If the START command is issued when no virtual machine is connected with an active path to \*MONITOR for event data, event recording is held pending until there is a connection with an active path.

Specific Usage Note for BLOCK:

1. The default BLOCK value is four 4 KB pages.
2. The minimum value for BLOCK is four 4 KB pages.
3. The maximum value for BLOCK is half the number of pages to be used for the event data area (partition-event config size).
4. The value of BLOCK is saved when monitoring is stopped.

Specific Usage Notes for PARTITION:

1. The default PARTITION size is half the size of the saved segment. The minimum is nine pages.
2. The minimum PARTITION size is eight pages plus the number of pages reserved for event configuration records.

### Notes:

- a. The QUERY MONITOR EVENT command can be used to display the number of pages currently designated for event configuration records.
- b. The MONITOR EVENT CONFIG SIZE command can be used to specify the number of pages to be reserved for the event configuration area. If the MONITOR EVENT CONFIG command has not been issued before the MONITOR START is issued, 68 pages are reserved for event configuration recording, and the PARTITION size must be at least 76 pages.
- c. The maximum PARTITION size depends on whether sample monitoring has already been started, and on whether any user is connected to \*MONITOR when the command is issued to start event monitoring.

If sample monitoring has *not* started, and:

- If one or more users are connected, the maximum PARTITION value that can be entered is the number of pages in the first type SC range of the saved segment that is currently being used by the monitor.
- If there are no users connected to \*MONITOR, the maximum value for PARTITION is equal to the maximum number of pages that can be specified as a type SC range with the DEFSEG command.

If sample monitoring **has** started, the maximum number of pages that can be specified by PARTITION is reduced by one more than the number of pages reserved for sample configuration records. (QUERY MONITOR EVENT and QUERY MONITOR SAMPLE display the number of pages currently reserved for the event and the sample configuration areas.)

3. The maximum PARTITION size depends on whether sample monitoring has already been started, and on whether any user is connected to \*MONITOR when the command is issued to start event monitoring.
4. When monitoring is stopped, the value of PARTITION reverts to the default value.

## Responses

Response 1:

```
MONITOR SAMPLE STARTED -- RECORDING IS ACTIVATED.
```

- or -

```
MONITOR SAMPLE STARTED BY [userid] -- RECORDING IS ACTIVATED.
```

As a result of the MONITOR START command, sample monitoring has successfully started. There is at least one user with an active path connected to \*MONITOR for sample data collection.



```
MONITOR SAMPLE STARTED -- RECORDING IS PENDING.
```

- or -

```
MONITOR SAMPLE STARTED BY [userid] -- RECORDING IS PENDING.
```

MONITOR START has been issued. Because there is no user with an active path connected to \*MONITOR for sample data collection, sample monitoring is held pending. It shall remain pending until the first user is connected to \*MONITOR for sample data collection.

```
MONITOR EVENT STARTED -- RECORDING IS ACTIVATED.
```

- or -

```
MONITOR EVENT STARTED BY [userid] -- RECORDING IS ACTIVATED.
```

As a result of the MONITOR START command, event monitoring has successfully started. There is at least one user with an active path connected to \*MONITOR for event data collection.

```
MONITOR EVENT STARTED -- RECORDING IS PENDING.
```

- or -

```
MONITOR EVENT STARTED BY [userid] -- RECORDING IS PENDING.
```

MONITOR START has been issued. Because there is no user with an active path connected to \*MONITOR for event data collection, event monitoring is held pending. It shall remain pending until the first user is connected to \*MONITOR for event data collection.

#### Notes:

1. Only one of the above responses is displayed for sample monitoring; another is displayed for event monitoring. Depending on the profile set up for monitoring, one or the other, or both, will be displayed.
2. These responses are sent to the user that issued the command. If this user is not the system operator, the responses containing [userid] are also sent to the system operator.

Response 2:

```
MONITOR SAMPLE STOPPED
```

- or -

```
MONITOR SAMPLE STOPPED BY [userid]
```

As a result of the MONITOR STOP command, sample monitoring has been stopped.

```
MONITOR EVENT STOPPED
```

- or -

```
MONITOR EVENT STOPPED BY [userid]
```

As a result of the MONITOR STOP command, event monitoring has been stopped.

#### Note:

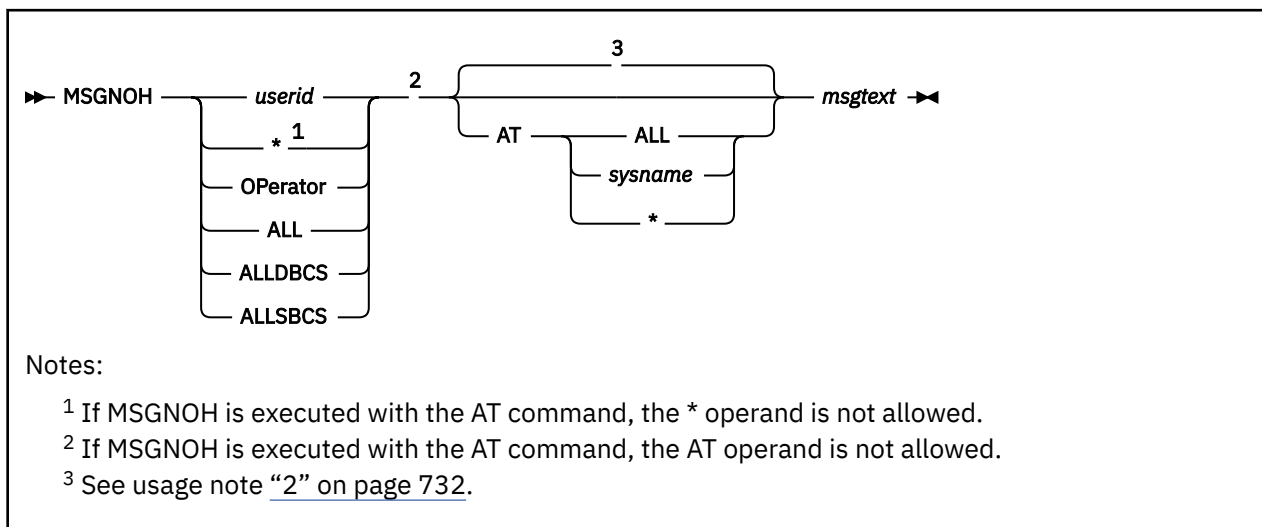
1. Only one of the above responses is displayed for sample monitoring; another is displayed for event monitoring. Depending on the profile set up for monitoring, one or the other, or both, will be displayed.
2. These responses are sent to the user that issued the command. If this user is not the system operator, the responses containing [userid] are also sent to the system operator.

## MONITOR START/STOP

### Messages

- HCP026E Operand missing or invalid
- HCP6227E Monitor {event|sample} collection is not active.

## MSGNOH



### Authorization

Privilege Class: B

### Purpose

Use MSGNOH to allow a service virtual machine to send messages without the standard MESSAGE command header.

### Operands

*userid*

\*

**OPERATOR**

**ALL**

identifies the user to whom you are sending the message. To send a message to yourself, specify an asterisk (\*).

Specify OPERATOR if you want to send a message to the primary system operator, regardless of the operator's user ID. If you specify ALL, you broadcast the message to all users receiving messages.

**ALLDBCS**

transmits the message to all users logged on to DBCS (double-byte character set) terminals.

**ALLSBCS**

transmits the message to all users logged on to SBCS (single-byte character set) terminals.

**AT ALL**

transmits the message to all systems in the SSI cluster to which this system belongs. The lowest return code is returned to the command issuer.

**AT *sysname***

transmits the message to a specific system in the SSI cluster to which this system belongs. The return code is returned to the command issuer.

**AT \***

transmits the message only to the system where the command is issued. This is the default if the system is not a member of an SSI cluster.

***msgtext***

is the text of the message you are sending. The length of the message is limited by the number of characters remaining on the input line after the command and appropriate operand are entered. DBCS text is supported as message text.

**Usage Notes**

1. If a message is issued from a CMS environment, the command and the message text cannot be longer than 240 characters.  
If the message is issued from a CP environment, the command and message length is limited by the input area of the terminal.
2. If the AT operand is not specified:
  - If MSGNOH is not executed with the AT command:
    - If the sender's system is not a member of an SSI cluster, the default is AT \*.
    - If the sender's system is a member of an SSI cluster, the default is AT ALL, except in the following cases, when the default is AT \*:
      - If the specified user ID is OPERATOR.
      - If the specified user ID is logged on to the same system as the sender.
      - If the specified user ID or the sender is defined by a multiconfiguration virtual machine definition.
      - If ALL, ALLDBCS, or ALLSBCS is specified, and a potential recipient user ID is defined by a multiconfiguration virtual machine definition, the default is AT \* for that user ID.
  - If MSGNOH is executed with the AT command, the AT operand function is bypassed (there is no default), and the message is sent to the system specified on the AT command.
3. If an external security manager (ESM) is installed on your system, you might not be authorized to enter this command. However, messages sent to or from the system operator and messages sent with the ALL option are not subject to authorization checking by the ESM. For additional information, contact your security administrator.
4. For additional information, see [“MESSAGE” on page 673](#).

**Responses**

Response 1:

```
msgtext
```

is the response received by the specified user or users.

***msgtext***

identifies the message text.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP057I *userid* not receiving; *text*
- HCP1142E The AT option is not allowed when this command is executed remotely with the AT command.
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.

- HCP6600E An error was detected by installation-wide {*CP module name*|exit point 1210} - return code *nnnn*

## NOTREADY

---

▶ NOTReady — *vdev* ▶

### Authorization

Privilege Class: G

### Purpose

Use NOTREADY to change the apparent status of a virtual spool or console from ready to not ready.

### Operands

#### *vdev*

identifies the number of the virtual device to be made not ready.

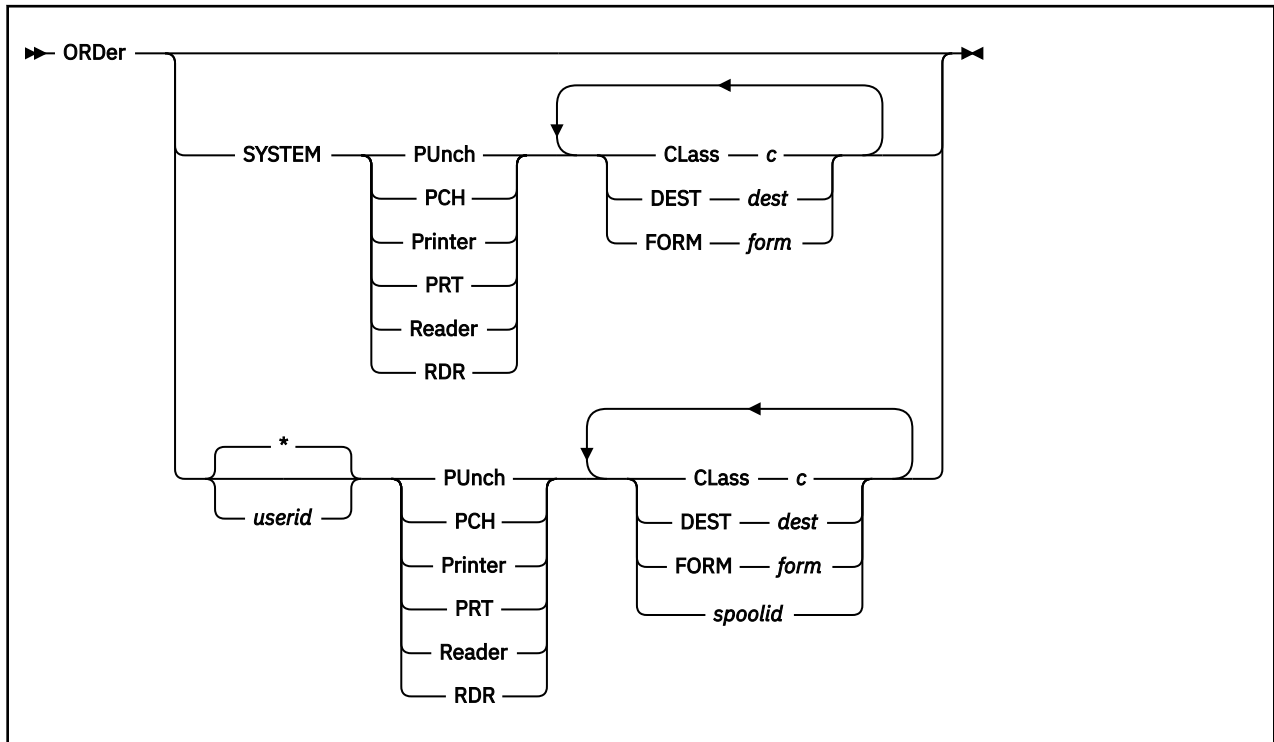
### Usage Notes

1. Use this command to change the status of spooled unit record devices and virtual consoles only.
2. Any I/O operation in progress to the specified device at the time the command is issued is completed; the not-ready condition is in effect for the next I/O operation.

### Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command.

## ORDER



### Authorization

Privilege Class: D, G

### Purpose

Use ORDER to change the order of a specified output queue.

Class G users can use ORDER to arrange their own spool files. Class D users can use ORDER to arrange any spool files in the system, regardless of owner.

### Operands

#### SYSTEM

is restricted to class D users. Use the SYSTEM operand to order any file or group of files on the system queue that you specify without indicating the owner's user ID. This is not valid with the *spoolid* option.

\*

#### userid

identifies the user whose spool files are to be ordered. If you are ordering your own files, you may specify *userid* as an asterisk (\*).

The *userid* option allows a class D user to order any file, or group of files, owned by a particular user. Reader files are repositioned only with respect to other reader files owned by the user ID that you specified. Print or punch files are repositioned to the top of the system queue and to the specified user output queue.

The *userid* option allows a class G user to order any file originated by that user, and that is now owned by the user whose user ID is specified on the command line. The files are only in relation to any other files that the class G user originated which are owned by the user ID specified in the command line.

## ORDER

**PUnch**  
**PCH**  
**Printer**  
**PRT**  
**Reader**  
**RDR**

indicates the spool file queue you want to order.

### **Class *c***

are the file classes that you want to rearrange, and the sequence in which they are to be processed. You must enter the keyword **Class** for each class that you want to order. The operand *c* is a 1-character alphanumeric fields (from A to Z or from 0 to 9) that represent spooling classes.

### **FORM *form***

are the forms to be rearranged, and the sequences in which they are to be processed. The operand *form* is specified as one to eight characters in length. For the class G user, the form is the user form name. For the class D user, the form is the user form name when you specify asterisk (\*) or default to your user ID. Otherwise, the form is the operator form number.

### **DEST *dest***

are the file destination values to be rearranged, and the sequence in which they are to be processed. The operand *dest* is specified as 1- to 8-character names.

### ***spoolid***

identifies the files you want to order. This option is not valid with the **SYSTEM** keyword.

## Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.
2. The order of closed spool files determines the sequence in which they are selected and processed.
3. Class G users can order only their own files and any files that they originated. The order that you specify does not affect the position of any other files that are in the same system queue as your files, but only repositions your files with respect to any other files you may own.

For example, suppose the system printer queue contains the following files:

<b><u>Spoolid</u></b>	<b><u>Owning User</u></b>
1	USER1
22	USER2
333	USER2
4444	USER2

Assume further that USER2 enters `order prt 4444`; the following queue order results:

<b><u>Spoolid</u></b>	<b><u>Owning User</u></b>
1	USER1
4444	USER2
22	USER2
333	USER2

The class G user may order files originated by that user and that another user ID now owns. If so, the user changes the order of the files only in relation to any other files that the user originated that are now owned by the other user ID. For example, assume there are two users, USER1 and USER2, and USER2's printer queue is as follows:



Spoolid	Owning User	Originator
123	USER2	USER2
456	USER2	USER1
789	USER2	USER1
9101	USER2	USER2

Assume further that USER1 enters `order user2 prt 789`; the following print queue order results for USER2:

Spoolid	Owning User	Originator
123	USER2	USER2
789	USER2	USER1
456	USER2	USER1
9101	USER2	USER2

4. The class D user can reorder any files in the system by specifying the SYSTEM or *userid* option.

For example, if you enter `order system printer class c`, all class C printer files are placed ahead of any other printer files in the system, regardless of who owns them or originated them.

The ORDER command is also useful to the class D user who wants to place files owned by a particular user ahead of all other files on a printer or punch queue. For example, if you enter `order user2 punch class b`, you place all of USER2's Class B punch files ahead of all other punch files on the system.

5. Use the QUERY READER/PRINTER/PUNCH command to determine spool file attributes, such as class and spool ID, for the files that you want to rearrange. The display produced by the command shows the current order of the spool files on the queue that you specify.
6. You can use a combination of CLASS, FORM, DEST and *spoolid* specifications to rearrange files. For example, `order printer class a 1963 class c form listing dest bldg2` orders the command issuer's printer queue as follows:
- All class A files
  - The file with spoolid 1963
  - All class C files
  - All files with the form LISTING
  - All files with the dest BLDG2.

They are followed by any other printer spool files not specified in the ORDER command line.

7. A file cannot be ordered more than once in a given command. For example, assume the following is a user's spool print queue.

Spoolid	Class
1234	A
4567	B
7890	A
5555	A
8888	B

If the user enters `order prt 8888 class a class b`, the following order results:

## ORDER

<u>Spoolid</u>	<u>Class</u>
8888	B
1234	A
7890	A
5555	A
4567	B

Note that *spoolid* 8888 was only ordered once.

8. Files that appear in PRINTED status when queried will not be selected for ORDER processing.

## Responses

Response 1:

```
{nnnnnn} FILE(S) ORDERED  
{      NO}
```

indicates the number of files ordered.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP008E Invalid spoolid - *spoolid*
- HCP026E Operand missing or invalid
- HCP028E Class missing or invalid
- HCP035E Device type missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*

## PER

---

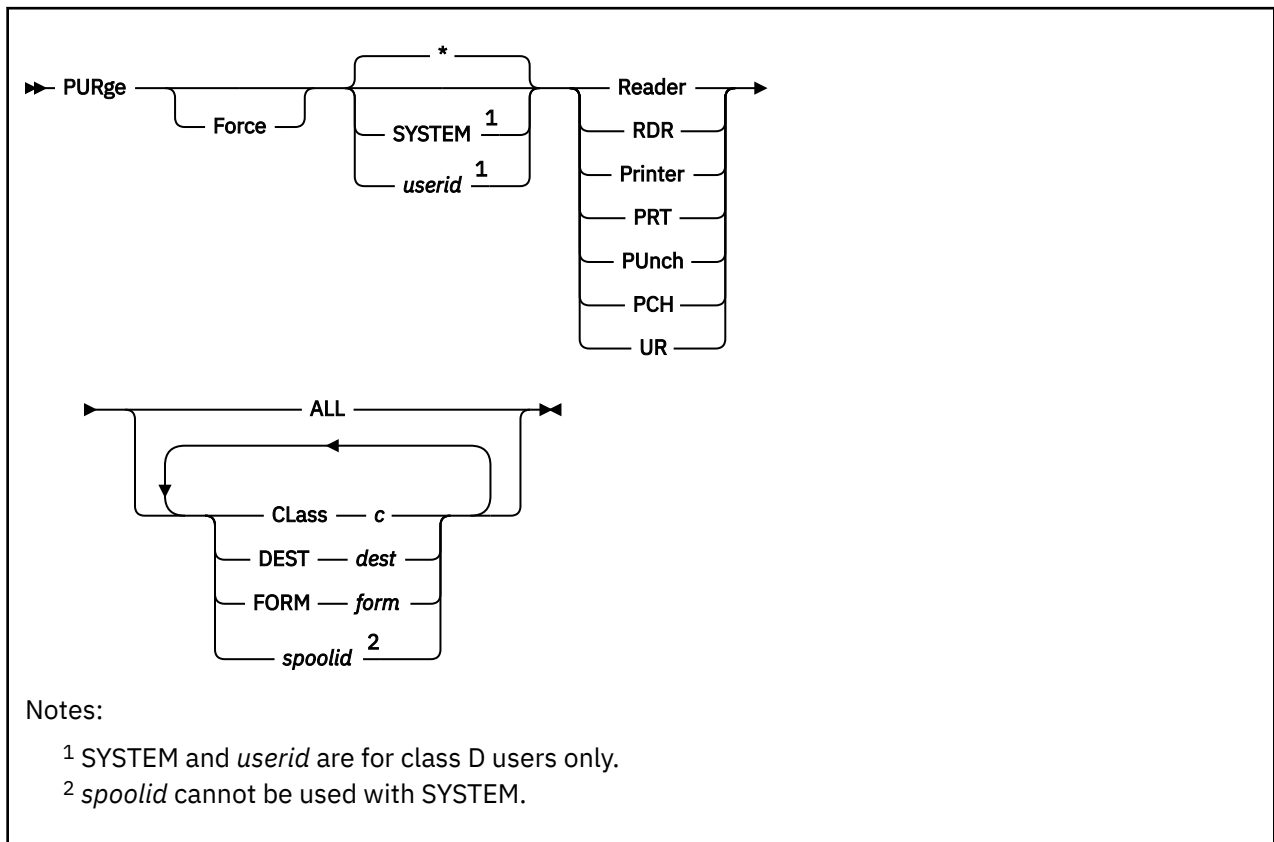
### **Authorization**

Privilege Class: G

### **Purpose**

PER is a synonym for TRACE. Within z/VM, TRACE can monitor events that occur in your virtual machine. For more information, see [“TRACE” on page 1946](#).

## PURGE



### Authorization

Privilege Class: D, G

### Purpose

Use PURGE to remove closed spool files from the system before they are selected for output.

PURGE *spoolid* can be used to purge a file after it begins to be processed by a real device; output function ceases and the spool file is purged. Class G users can use PURGE to purge their own spool files. Class D users can use PURGE to purge any spool files in the system,

### Operands

#### Force

indicates that the files matching the selection criteria are purged immediately by CP, even files that have been converted by the print server. If FORCE is not specified, and a file has been converted, CP releases all of the file's resources, except for the file descriptor block, and marks the file as logically purged. When the print server selects a file for processing, it acknowledges that the file has been purged and tells CP to release the file descriptor. FORCE is valid for the Class D user only.

#### SYSTEM

allows the class D user to manipulate all files in the system, regardless of ownership. This is not allowed with *spoolid*.

\*

#### *userid*

identifies the user whose files are to be purged.

The operand *userid* allows the class D user to purge files that belong to any user on the system. If you are a class G user, you can purge your own files or files owned by another user ID that you originated.

Class D or class G users can purge their own files using the asterisk (\*) option. If you do not specify *userid* or \*, the default is to purge your own files.

**Reader****RDR****Printer****PRT****PUnch****PCH****UR**

indicates which spool file queue contains the files that you want to purge. UR indicates that all reader, printer, and punch files are to be purged.

**ALL**

specifies that all files on the queue of the device type specified (READER, PRINTER, PUNCH, UR) are to be purged.

**Class c**

designates a spool file class where *c* is a 1-character alphanumeric field from A to Z or 0 to 9. All the files of the specified class or classes for the specified device type or types are purged.

**DEST *dest***

purges all of the files with the specified destination value. The operand *dest* is specified as one to eight characters long.

**FORM *form***

purges all files with the specified form names or form numbers. The operand *form* is specified as one to eight characters long. For the class G user, the form is the user form name. For the class D user, the form is the user form name when you specify \* or default to your *userid*. Otherwise, the form is the operator form number.

***spoolid***

purges the files of the specified *spoolid*, for the specified device type.

**Usage Notes**

1. SET IMSG OFF suppresses informational messages for this command.
2. The class D user can use the PURGE command to purge any file in the system by specifying the SYSTEM or *userid* option. For example, if you enter `purge system printer class c`, you purge all class C printer files in the system, regardless of who owns them or who originated them.  
  
The class D user can use the *userid* option to purge any file or files that are owned by a particular user. For example, if you enter `purge genuser reader all`, you purge all reader files owned by the user ID GENUSER.
3. You can purge files using a combination of CLASS, FORM, DEST, and *spoolid* on the same command line. For example, `purge printer class a 1932 form document class d 619 dest bldg1` specifies that CP should purge all the command user's class A and class D printer files, all the command user's files with form DOCUMENT, all the command user's printer files with the destination BLDG1, and the command user's printer files having *spoolids* 1932 and 619.
4. If a command syntax error occurs when purging closed spool files from the system, only the files encountered prior to the error are processed.

**Responses**

Response 1:

```
{nnnnnn} FILE(S) PURGED
{
  NO}
```

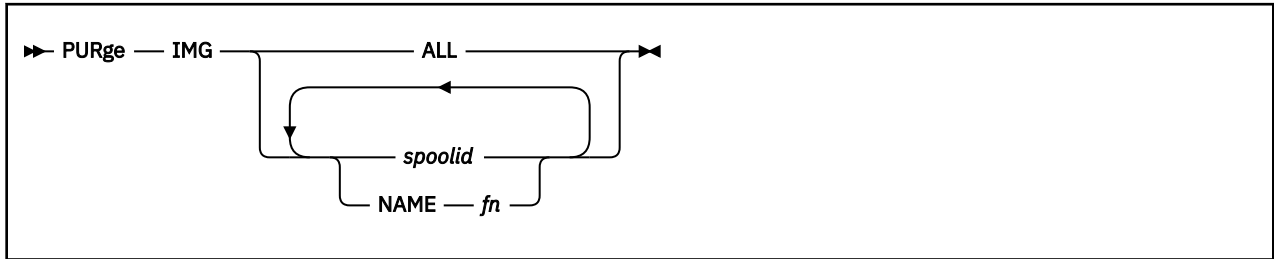
## PURGE

indicates the number of files purged.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP008E Invalid spoolid - *spoolid*
- HCP026E Operand missing or invalid
- HCP028E Class missing or invalid
- HCP035E Device type missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP356E Access denied; User *userid* file *spoolid* not {changed|transferred|printed}
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP644E Unknown/unauthorized CP command: *command*
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP1001E An operand is missing for *option*.
- HCP1012E Queue type missing or invalid
- HCP1013E An invalid operand was supplied for *option* - *operand*
- HCP1370E You cannot purge *name* using the ASSOCIATES option.
- HCP6525E Authorization request failed.

## PURGE IMG



### Authorization

Privilege Class: A, B, C, D, E

### Purpose

Use PURGE IMG to remove system data files containing image libraries.

### Operands

#### ALL

lets you purge all image library files.

#### *spoolid*

identifies the number used to identify the file or files to be purged.

#### NAME *fn*

identifies the file name of the file or files to be purged.

### Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.
2. Any combination of *spoolid* and NAME may be specified on the PURGE IMG command line. For example, `purge img 1234 name imag3800 3421` purges IMG files 1234 and 3421, and any IMG files with the file name IMAG3800.
3. In response to the PURGE command, CP purges the designated files *unless* the files are currently in use. If a file is currently in use, CP places it in a pending purge state (class P) and purges it as soon as it is no longer being used or the system is shut down or re-IPLed.
4. To determine if a file is in a pending purge state, enter the QUERY IMG command. If the file *is* in a pending purge state, CP's response shows that the file is Class P.

### Responses

Response 1:

```
{nnnn} FILE(S) PURGED
{ NO }
```

displays the count of files purged.

#### *nnnn*

indicates the number of files purged.

Response 2:

```
{nnnn} FILE(S) PENDING PURGE
```

## PURGE IMG

displays the count of files pending purge.

### ***nnnn***

indicates the number of files to be purged when they are no longer in use.

For example, in the situation where five image files exist, of which two are in use, enter `purge img all`. The responses are:

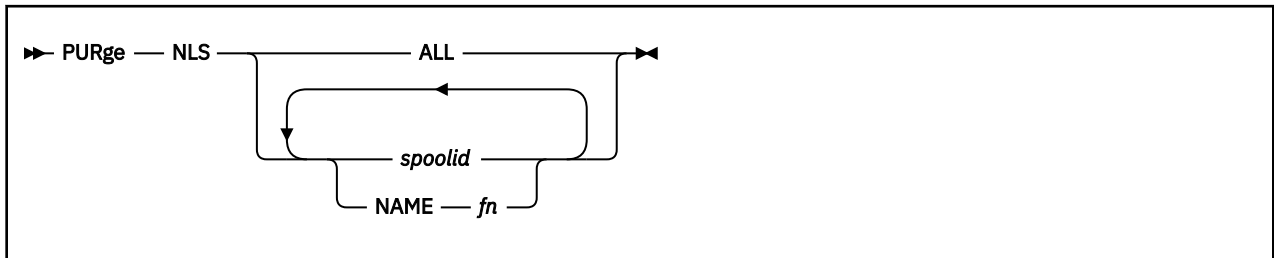
```
0003 FILES PURGED
0002 FILES PENDING PURGE
```

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP008E Invalid spoolid - *spoolid*
- HCP026E Operand missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*



## PURGE NLS



### Authorization

Privilege Class: E

### Purpose

Use PURGE NLS to purge message repository files.

### Operands

#### ALL

specifies that all message repository files are to be purged.

#### *spoolid*

identifies the file to be purged.

#### NAME *fn*

specifies the name of the file to be purged.

### Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.
2. Any combination of *spoolid* and NAME may be specified on the PURGE NLS command line. For example, `purge nls 1234 name nls101 3421` purges message repository files 1234 and 3421, and any message repository files with the file name NLS101.

### Responses

Response 1:

```
{nnnn} FILE(S) PURGED
{ NO}
```

displays the count of files purged.

#### *nnnn*

indicates the number of files purged.

Response 2:

```
{nnnn} FILE(S) PENDING PURGE
```

displays the count of files waiting to be purged.

#### *nnnn*

indicates the number of files to be purged when they are no longer in use.

## PURGE NLS

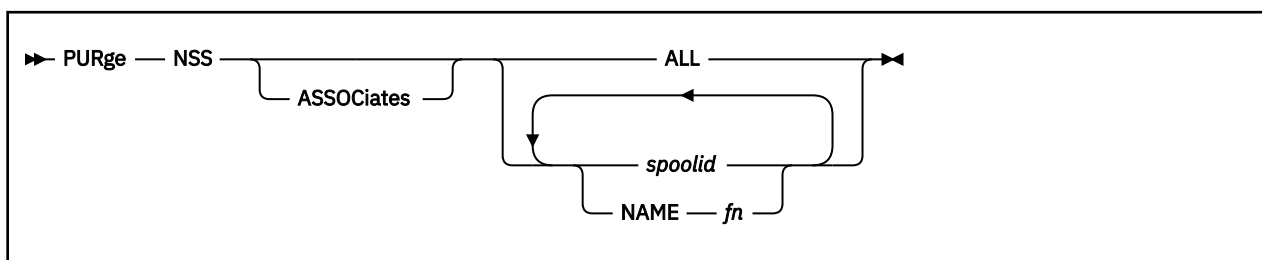
For example, when five NLS files exist and two are in use, and you enter `purge nls all`, the responses are:

```
0003 FILES PURGED
0002 FILES PENDING PURGE
```

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP008E Invalid spoolid - *spoolid*
- HCP026E Operand missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP423E The function requested is not compatible with a previously specified operand.
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*

## PURGE NSS



### Authorization

Privilege Class: E

### Purpose

Use PURGE NSS to remove system data files containing named saved systems (NSSs) and saved segments.

### Operands

#### ASSOCIates

specifies that the files associated with a segment space or member segment are to be purged. If you purge a segment space, its associated members are also purged or set to pending purge, or their segment space entries are compressed. If a segment space is purged with ASSOCIATES, then all the members in that space, that are not defined to another segment space (same ranged), are purged. If the member(s) belong to another segment space they will not be purged.

If you purge a member segment, it is purged or set to pending purge, and its associated segment spaces are purged or set to pending purge, or their member entries are compressed. If a member of a segment space is purged with ASSOCIATES, then the member will be purged when the use count=0, and the member will be removed from all of the segment spaces that it is defined to. Once the use count=0, and the member is purged, then the segment spaces it was previously defined to will complete and can be loaded from. If associates option is not used, the member will be purged when the use count=0, but the space will no longer be loadable.

#### ALL

lets you purge all NSS and saved segment files.

#### *spoolid*

is the number used to identify the file. It specifies the file to be purged. It can be an NSS, a saved segment, a segment space, or a member saved segment.

#### NAME *fn*

specifies the file name that identifies the file to be purged. The file can be an NSS, a saved segment, a segment space, or a member saved segment.

### Usage Notes

1. SET IMMSG OFF suppresses informational messages for this command.
2. Any combination of *spoolid* and NAME may be specified on the PURGE NSS command line. For example, `purge nss 1234 name cms 3421` purges NSS and saved segment files 1234 and 3421, and any NSS and saved segment files with the file name CMS.
3. The PURGE NSS command interrupts an in-process SAVESEG or SAVESYS operation to purge the saved segment or named saved system before the save completes. This will not leave a partially-saved

## PURGE NSS

system data file on the system. The skeleton system data file will be purged and must be redefined before the saved segment or named saved system can be resaved.

4. In response to the PURGE command, CP purges the designated files *unless* the files are currently in use. If a file is currently in use, CP places it in a pending purge state (class P), and purges it as soon as it is no longer being used or the system is shut down or re-IPLed. If the file being purged is a member of a segment space, the system places it in pending purge status if any user is accessing any member of the segment space. The system purges the member when the segment space is no longer in use.
5. If you do not specify ASSOCIATES and the target of the PURGE command is a member saved segment, CP purges only that member. It does not purge any other file, and it does not update any segment space descriptions. If the target of the PURGE command is a saved segment, CP purges only the saved segment. If the target of the PURGE command is a segment space, CP purges only that segment space. It does not purge any other file, and it does not update members of that segment space.
6. To determine if a file is in a pending purge state, enter the QUERY NSS command. If the file is in a pending purge state, the CP response shows that the file as class P.
7. Processing of a PURGE NSS command might be delayed if DIAGNOSE code X'64' is concurrently performing the initial load of a large DCSS.

## Responses

Response 1:

```
{nnnn} FILE(S) PURGED  
{ NO}
```

displays the count of files purged.

### **nnnn**

indicates the number of files purged.

Response 2:

```
{nnnn} FILE(S) PENDING PURGE
```

displays the count of files waiting to be purged.

### **nnnn**

indicates the number of files to be purged when they are no longer in use.

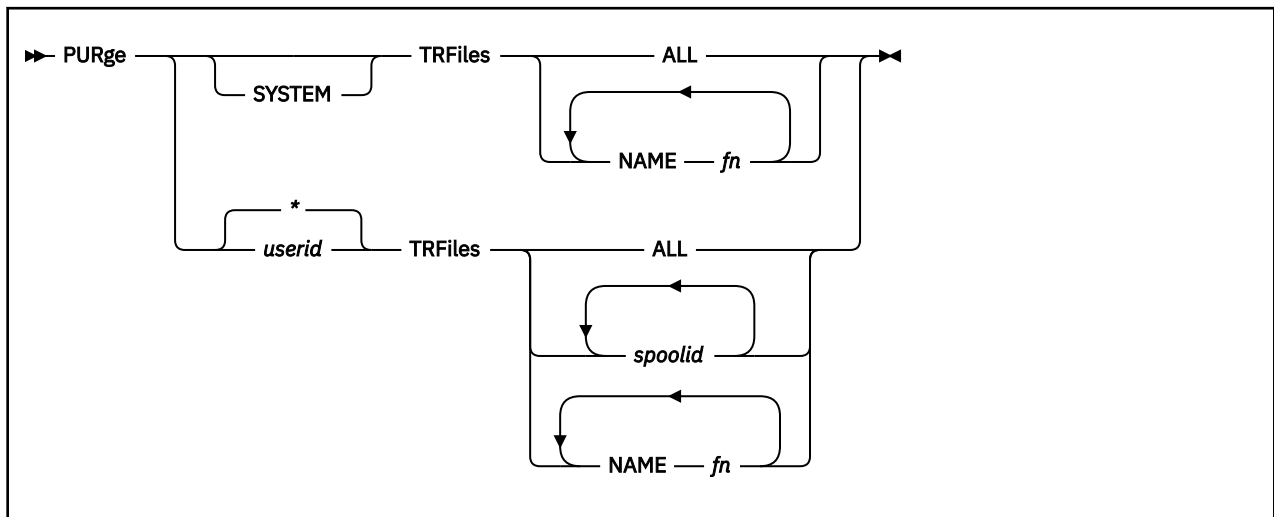
For example, when five NSS or saved segment files exist and two are in use, and you enter `purge nss all`, the responses are:

```
0003 FILES PURGED  
0002 FILES PENDING PURGE
```

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP008E Invalid spoolid - *spoolid*
- HCP026E Operand missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*
- HCP1370E You cannot purge *name* using the ASSOCIATES option.

## PURGE TRFILES



### Authorization

Privilege Class: A, C, D, E, G

### Purpose

Use PURGE TRFILES to remove system trace files created by TRSAVE.

### Operands

#### SYSTEM

specifies that files are to be purged regardless of who owns or originates them. Class G users cannot use the SYSTEM option; it is not valid with the *spoolid* option.

\*

#### *userid*

specifies the owner of the files to be purged. System trace files cannot be purged while they are in use. Class G users can purge only files that they own or originate, and that are not in use. Class A, C, D, and E users can purge any system trace files that are not in use.

You can specify asterisk (\*) to purge your own system trace files. This is the default.

#### TRFiles

specifies that system trace files are to be purged.

#### ALL

specifies that all system trace files are to be purged.

#### NAME *fn*

specifies the name of the file to be purged. This is the file name specified on the command TRSAVE... ON DASD NAME *fname*... when the trace was started. More than one file can exist with the same file name for a given owner.

#### *spoolid*

specifies the spool file identification number of the file to be purged. This option is not valid with the SYSTEM option.

### Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.

## PURGE TRFILES

2. Any combination of *spoolid* and NAME may be specified on the PURGE TRFILES command line. For example, `purge trf 1234 name gtrace 3421` purges system trace files 1234 and 3421, and any trace files with the file name GTRACE.
3. If you specify *spoolid* to purge files, the file you purge may be part of a group of files with the same file name as that created by the TRSAVE...ON DASD NAME *fname*... command. If this is the case, some system trace data information is purged.
4. If you specify the NAME option to purge files and there are multiple groups of trace files with the same name, you may accidentally purge files you do not want to purge. Use the QUERY TRFILES command to display all of the trace files and determine which files you want to discard. If more than one file exists with the same name, use *spoolid* to specify the correct file to be purged and to avoid losing data.
5. If you are authorized to use the SYSTEM option and you use it with the NAME option, you purge all trace files in the system with the file name specified.
6. If a system trace file has a class of W (meaning the file is being written to), it is not purged. You must enter the TRSAVE OFF, TRSAVE CANCEL, or TRSOURCE DISABLE command to purge the class W file.

## Responses

Response 1:

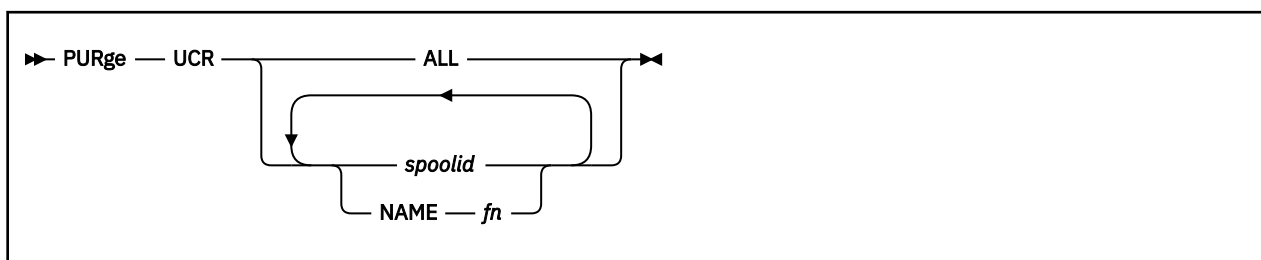
```
{nnnnnnn} FILE(S) PURGED  
{          NO}
```

indicates the number of files purged.

## Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*}
- HCP008E Invalid spoolid - *spoolid*
- HCP026E Operand missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*

## PURGE UCR



### Authorization

Privilege Class: A, B, C

### Purpose

Use PURGE UCR to purge a UCR file created by the system programmer.

### Operands

#### ALL

lets you purge all UCR files.

#### *spoolid*

is the number used to identify the files to be purged.

#### NAME *fn*

is the file name used to identify the files to be purged.

### Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.
2. You may specify any combination of *spoolid* and NAME on the PURGE UCR command line. For example, `purge ucr 1234 name ucr101 3421` purges UCR files 1234 and 3421, and any UCR files with the file name UCR101.
3. UCR spool files and the OVERRIDE command are no longer supported. Once these files are purged, they cannot be rebuilt. Use the MODIFY statement or command to change command privilege classes. For information about MODIFY, see [Modifying a Command and Diagnose Code in z/VM: CP Exit Customization](#).

### Responses

Response 1:

```
{nnnn} FILE(S) PURGED
{ NO}
```

displays the count of files purged by the PURGE UCR command.

#### *nnnn*

indicates the number of files purged.

### Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP008E Invalid spoolid - *spoolid*

## PURGE UCR

- HCP026E Operand missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*



## QUERY ABEND

► Query — ABEND ◄

### Authorization

Privilege Class: B

### Purpose

Use QUERY ABEND to display the status of the soft abend function. Using this command, you can determine whether soft abends cause full system termination, full system snapdumps, or are processed normally.

### Responses

Response 1:

```

ABEND  {SOFT      }
       {SOFT aaannn}
       {HARD      }
       {HARD aaannn}
       {OFF       }
       {OFF aaannn}
       {SNAPDUMP  }
       {SNAPDUMP aaannn}

```

#### SOFT

indicates that soft abends do not force full system termination; normal system processing continues after recovery action is taken.

#### SOFT *aaannn*

specifies the soft abends that you have selected to be treated as soft abends; normal system processing continues after recovery action is taken. The code *aaannn* is made up of a 3-character module name and a 3-character abend code.

**Note:** ABEND SOFT *aaannn* can display up to five entries on a single output response line.

#### HARD

indicates that no soft abends are to attempt recovery action; all soft abends cause full system termination as if they were hard abends.

#### HARD *aaannn*

specifies the soft abends that you have selected to be treated as hard abends; they cause full system termination. The code *aaannn* is made up of a 3-character module name and a 3-character abend code.

**Note:** ABEND HARD *aaannn* can display up to five entries on a single output response line.

#### OFF

indicates that soft abends will not generate a soft abend dump; normal system processing continues after recovery action is taken.

#### OFF *aaannn*

specifies the soft abends that you have selected will not generate a soft abend dump; normal system processing continues after recovery action is taken. The code *aaannn* is made up of a 3-character module name and a 3-character abend code.

**Note:** ABEND OFF *aaannn* can display up to five entries on a single output response line.

#### SNAPDUMP

indicates that all soft abends cause full system snapdumps, as if they were specified as snapdumps.

## QUERY ABEND

### SNAPDUMP *aaannn*

specifies the soft abends that you have selected to be treated as snapdumps; they cause snapdump dumps to be created. The code *aaannn* is made up of a 3-character module name and a 3-character abend code.

**Note:** ABEND SNAPDUMP *aaannn* can display up to five entries on a single output response line.

### Examples

```
ABEND SOFT
ABEND HARD      ABC001 BCD002 CDE003
ABEND SNAPDUMP DEF004 EFG005 FGH006 GHI007 HIJ008
ABEND SNAPDUMP IJK009
```

This would indicate that there were three soft abends that had been set to hard (*ABC001*, *BCD002* and *CDE003*), and six soft abends that had been set to snapdump (*DEF004*, *EFG005*, *FGH006*, *GHI007*, *HIJ008* and *IJK009*). The global setting is soft, which is in effect for all of the remaining soft abend codes.

### Messages

- HCP002E Invalid operand - *operand*

## QUERY ACCOUNT

---

► Query — ACCount ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY ACCOUNT to display the account number being charged for your session activity.

### Responses

Response 1:

```
ACCOUNT acntnum
```

#### ***acntnum***

identifies your current session account number.

### Messages

- HCP002E Invalid operand - *operand*

## QUERY ADJUNCT

---

► Query — ADJunct ◄

### Authorization

Privilege Class: G

### Purpose

Use to display the status of an adjunct machine for the user.

### Responses

Response 1:

When the adjunct configuration of userid VTEST1 is active (and thus the principal configuration is suspended):

```
HCPYJT5777I VTEST1 Adjunct is currently active
```

In this state, commands prefixed with #CP are directed by default to the adjunct configuration.

Response 2:

When the adjunct configuration of userid VTEST1 is suspended (and thus the principal configuration is active):

```
HCPYJT5776I VTEST1 Adjunct is suspended
```

In this state, commands prefixed with #CP are directed by default to the principal configuration.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP643I Adjunct feature is not enabled on this system
- HCP5768I *userid* Adjunct does not exist
- HCP5770I *userid* Adjunct is not exploitable
- HCP5776I *userid* Adjunct is suspended
- HCP5777I *userid* Adjunct is currently active

# QUERY AGELIST

► Query — AGELIST ◄

## Authorization

Privilege Class: A, E

## Purpose

Use QUERY AGELIST to display the settings for the global aging list used by the system's frame replenishment algorithm.

To satisfy the system's need for real storage, the frame replenishment algorithm selects candidate frames from the user and system frame-owned lists. The pages associated with the selected frames are put in the invalid-but-resident (IBR) state. Any access to a page in the IBR state by the guest results in a re-validation. Otherwise, after a period of aging on the local frame-owned list, the frame associated with an IBR page is added to the global aging list.

If a page is referenced while its frame is on the aging list, the page is re-validated and the frame moves back to the appropriate frame-owned list. Before a frame can be reclaimed from the aging list, the page content must be backed in auxiliary storage.

## Responses

```
Target size      = nU (mU) x.x% of pageable storage
In use          = nU
Pending writes  = nU
Early writes    = Yes|No
Sizing          = Fixed|Variable
Keep slot       = Yes|No
```

where:

### Target size *nU*

identifies the maximum expected amount of pageable storage to be contained in the global aging list. The target size is displayed in a storage unit (K,M,G,T,P,E) suitable to represent the values displayed in the remaining lines of the command response.

### (*mU*)

the target size in a storage unit with a precision of no more than 7 significant digits. This value may be rounded if the aging list size is defined as a percentage of pageable storage. When the aging list is empty this value is omitted.

### *x.x%* of pageable storage

identifies the target size as a percentage of pageable storage. The value may be rounded to the nearest tenth of a percent if the aging list size is defined as a fixed storage size.

### In use *nU*

identifies the amount of storage currently within the global aging list in storage units (K,M,G,T,P,E). Usage of the aging list is at the discretion of CP based on storage constraints. The amount of storage currently within the aging list may be significantly less than the target size, or in some cases slightly higher.

### Pending writes *nU*

identifies the portion of the global aging list in storage units (K,M,G,T,P,E), that must be backed up in auxiliary before the associated frames can be reclaimed.

## QUERY AGELIST

### **Early writes = Yes**

indicates the frame replenishment algorithm backs up page content to auxiliary storage in advance of frame reclaim to maintain a pool of immediately reclaimable frames.

### **Early writes = No**

indicates the frame replenishment algorithm backs up page content to auxiliary storage only when the system is in need of frames.

### **Sizing = Fixed**

indicates the size of the aging list was specified on the SET AGELIST command as a fixed storage size.

### **Sizing = Variable**

indicates the size of the aging list was specified on the SET AGELIST command as a percentage of pageable storage. Since the number of real frames containing pageable content varies over time, so will the size of the aging list.

### **Keep slot = Yes**

indicates the auxiliary storage address (ASA) to which a page is written during frame replenishment will remain allocated to that page when it is read into real memory upon access by the guest.

### **Keep slot = No**

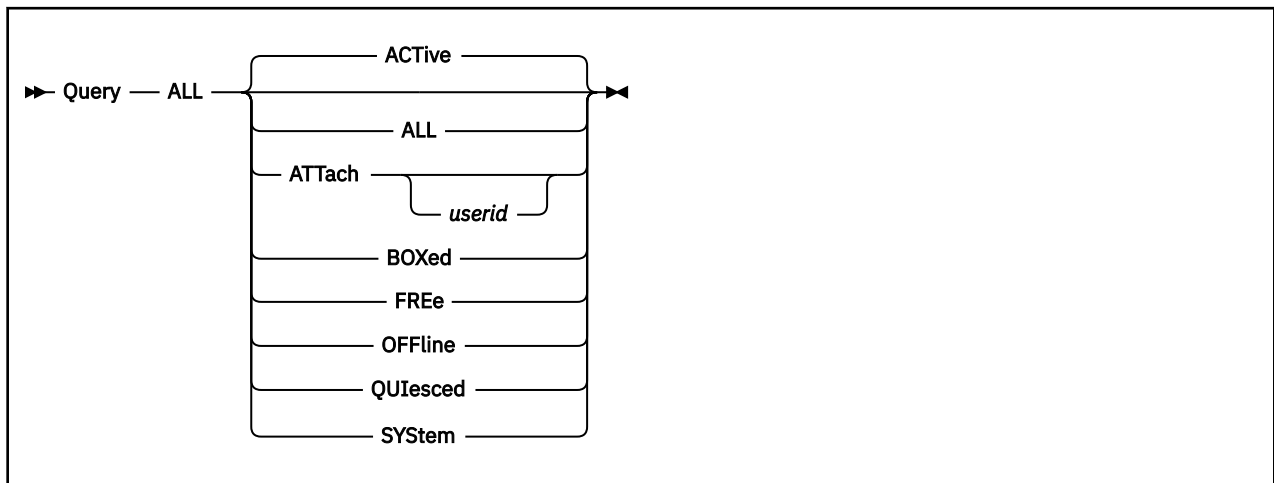
indicates the ASA to which a page is written during frame replenishment is unallocated during page fault processing.

When Keep slot is No, some resident pages might still have copies on auxiliary storage, unless Early writes is also No.

## **Messages**

- HCP003E Invalid option - *option*|command contains extra option(s) starting with *option*

## QUERY ALL



### Authorization

Privilege Class: B

### Purpose

Use `QUERY ALL` to display the status of all the real devices attached to the system. It also displays the size of real storage.

### Operands

#### ACTive

indicates all active devices. An active device is one that is in use by a user or the system and is specified as neither `FREE` nor `OFFLINE`.

#### ALL

indicates all devices attached to the system. The status is displayed in the order of `ACTIVE`, `FREE`, and `OFFLINE`. You receive the same response as you would receive if you entered all of the following commands:

```
query all active
query all free
query all offline
```

#### ATTach *userid*

indicates the devices that are dedicated to a user on the system. If you specify *userid*, only devices attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all devices dedicated to any user on the system. Attached devices are always active.

#### BOXed

indicates the devices that are unavailable (usually because of a hardware condition). A boxed device may also be active, free, or offline.

#### FREe

indicates all free devices. A free device is one that is not currently in use by a user or the system and is specified as neither `ACTIVE` nor `OFFLINE`.

#### OFFline

indicates all offline devices. An offline device is specified as neither `ACTIVE` nor `FREE`. The offline response will also show offline devices which are quiesced.

## QUIesced

indicated all quiesced DASD. A quiesced device is an offline device which cannot be varied online until a resume request is received for the device.

## SYStem

indicates the devices in use by the system. A system device is always active.

## Usage Notes

1. When you enter a QUERY ALL command, you must make sure the command's options are valid for each virtual machine associated with one or more *prtids*. For example, if you enter a QUERY ALL command with the ALL parameter and a service virtual machine (SVM) controlling a printer does not accept that parameter, you get an error message from the service virtual machine.
2. A VIRTUAL version of this command, QUERY VIRTUAL ALL, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL ALL” on page 1273](#).

## Responses

The response to a QUERY ALL command is a mixture of all of the following responses. CP presents the information you request in ascending order of real device numbers.

Response about Real Storage:

At the beginning of the response to any QUERY ALL command, you receive a line like this:

```
STORAGE = 0128768K
```

This means the system's real storage is approximately 128 megabytes.

Response for Devices Needing Intervention:

If the system is using any device that is in an intervention-required state and you enter QUERY ALL, you receive a response like this:

```
DASD 1801 has intervention required.
```

The message indicates the real device type and device number. In this case, the real device type is DASD, which includes emulated DASD. The message can include the following device types:

<b>Type</b>	<b>Meaning</b>
FCP	SCSI device adapter
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
LINE	Communication line
CTCA	Channel-to-channel adapter
OSA	Open Systems Adapter
CTLR	3745 Communications Controller
MSC	Mass Storage Controller
SWCH	Dynamic Switching device
DEV	Any other device.



**Note:** This response will be preceded by a combination of messages that look like the following.

Responses from Real Channel-to-Channel Adapters:

If there are channel-to-channel adapters attached to the system, you see something like this:

```
CTCA 0480 ATTACHED TO PVM      0480
CTCA 0481 ATTACHED TO PVM      0481
CTCA 0482 ATTACHED TO RSCS     0482
CTCA 0483 ATTACHED TO RSCS     0483
CTCA 0484 ATTACHED TO PVM      0485
CTCA 0485 ATTACHED TO PVM      0484
CTCA 0486 ATTACHED TO VTAM     0486
CTCA 0700 ATTACHED TO TCPIP    0500
CTCA 0701 ATTACHED TO TCPIP    0501
```

The first number (for example, 0480) is the real device number of the channel-to-channel adapter. The ATTACHED operand might also read BOX/ATTC, which means that although the device is attached and dedicated to a user, it is unavailable. PVM, RSCS, TCPIP, and VMNET are user IDs to which the CTCAs are dedicated. The last number (for example, 0480) is the virtual device number that each virtual machine uses to refer to a CTCA.

If there are free channel-to-channel adapters attached to the system, you see something like this:

```
CTCA 0487 FREE      , CTCA 04A1 FREE      , CTCA 04A2 FREE      , CTCA 04A3 FREE
CTCA 04A4 FREE      , CTCA 04A5 FREE      , CTCA 04A9 FREE      , CTCA 04AA FREE
CTCA 04AB FREE      , CTCA 04BC FREE      , CTCA 04BD FREE      , CTCA 04BE FREE
CTCA 04BF FREE
```

This response shows the real device number for each free CTCA.

The responses for offline and boxed adapters are in the same format.

Responses from FCP Adapter Devices:

If there are FCP devices attached to the system, you see something like this:

```
FCP 0480 ATTACHED TO LINUX1    0480 CHPID F1
FCP 0481 ATTACHED TO LINUX1    0481 CHPID F1
FCP 0482 ATTACHED TO LINUX2    0482 CHPID F1
FCP 0483 ATTACHED TO LINUX3    0483 CHPID F1
FCP 0484 ATTACHED TO TEST1     0485 CHPID F1
FCP 0485 ATTACHED TO TEST1     0484 CHPID F1
```

The first number is the real device number of the FCP adapter device. The ATTACHED operand might also read BOX/ATTC, which means that although the device is attached and dedicated to a user, it is unavailable. LINUX1, LINUX2, LINUX3, and TEST1 are user IDs to which the FCP devices are dedicated. The next number is the virtual device number that each virtual machine uses to refer to an FCP device. The last number is the CHPID number (channel path) associated with the device number.

If there are free FCP devices attached to the system, you see something like this:

```
FCP 04A1 FREE      , FCP 04A2 FREE      , FCP 04A3 FREE      , FCP 04A4 FREE
FCP 04A5 FREE      , FCP 04A6 FREE      , FCP 04A9 FREE      , FCP 04AA FREE
FCP 04AB FREE      , FCP 04BC FREE      , FCP 04BD FREE      , FCP 04BE FREE
FCP 04BF FREE
```

This response shows the real device number for each free FCP device.

The responses for offline and boxed devices are in the same format.

Responses from Real DASD:

You see responses like the following if real DASDs are attached to the system for access as user minidisks or for system paging, spooling, or temporary disk space.

```
DASD 0280 CP SYSTEM XAUSR1    49
DASD 0281 CP SYSTEM XAUSR2    24
DASD 0282 CP SYSTEM XAUSR3    49
DASD 0283 CP SYSTEM XAUSR4    34
DASD 0284 CP SYSTEM XAUSR5    42
DASD 0285 CP SYSTEM XAUSR6    20
```

## QUERY ALL

```
DASD 0A01 CP OWNED ESAP01 3
DASD 0A02 CP OWNED ESAP02 1
DASD 0A03 CP OWNED ESAP03 1
DASD 0A04 CP OWNED ESAP04 1
DASD 0A05 CP OWNED ESAP05 1
DASD 0C08 CP SYSTEM ESAP7 1
DASD 0C09 CP SYSTEM ESAP6 1
DASD 1800 CP SYSTEM USP1 4
DASD 1801 CP SYSTEM USP2 2
```

The first number listed (for example, 0280) is the real device number of the DASD. SYSTEM indicates the device is allocated to the system for use as users' minidisks. OWNED indicates the device is used by the system for paging and spooling activity. XAUSR1, ESAP01, and the rest are volume labels of the devices. The last number (for example, 49) indicates the number of links to minidisks contained on the volume.

The following response is also displayed if real DASDs are attached to the system for access as user minidisks or for system paging, spooling, or temporary disk space. This response is longer than the response above because the device may be reserved (shared option is on), it may have a reserve pending, or it is currently reserved.

```
DASD 181A CP SYSTEM ESAP2C SHARED
DASD 181B CP OWNED ESAP2F RESERVE PENDING FOR USER DAVID
DASD 181C CP OWNED ESAP30 RESERVED BY SYSTEM
```

If a DASD is dedicated to a user, you see a message like this:

```
DASD 0820 ATTACHED TO OLIVER 01FC R/O XAUSR4
```

This means that the DASD at the real device number 0820 is attached to OLIVER's virtual machine as the virtual device 01FC. The DASD's volume identifier is XAUSR4. R/O means the user OLIVER has read-only access to the DASD. R/O might also read R/W, which means the user has read-write access. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

Responses from Real Graphic Display Devices:

If a real graphic display is dedicated to a virtual machine, you see something like this:

```
GRAF 0BCD ATTACHED TO OLIVER 0010
```

This message means the graphic display device at the real device number 0BCD is attached to OLIVER's virtual machine as the virtual device 0010. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

If a graphic display device has a user logged on or dialed through it, you see something like this:

```
GRAF 08E0 LOGON AS SVCDIR 0009
VTAM CONTROLS THE FOLLOWING LU'S
GRAF WGU9E316 LOGON AS DONNA 0009
GRAF TN6PV192 LOGON AS CORAK 0009
```

This response tells you the user SVCDIR is logged on through the graphic display device at 08E0. The number 0009 is SVCDIR's virtual console device number.

The second line of the response tells you the terminals are SNA terminals and the user ID VTAM controls them. VTAM is a VTAM service machine (VSM) recognized by z/VM CP. Any logical units controlled by this VTAM service machine that are logged onto or dialed into the system are described in the response lines following this line.

The numbers WGU9E316 and TN6PV192 identify SNA terminals controlled by a VTAM service machine. They are the names from the VTAM definition of the terminals as SNA logical units.

DONNA and CORAK are the users logged on through VTAM; each has 0009 as the virtual console device number.

If there are free graphic display devices, you see something like this:

```
LINE 00E5 FREE , LINE 00E6 FREE , LINE 00C0 FREE
GRAF 08E1 ENABLED , GRAF 08E2 ENABLED , GRAF 08E3 ENABLED , GRAF 08E4 ENABLED
GRAF 08E5 ENABLED , GRAF 08E6 ENABLED , PRT 08E7 ENABLED , GRAF 0BC0 ENABLED
GRAF 0BC1 ENABLED , GRAF 0BC2 ENABLED , GRAF 0BC3 ENABLED , GRAF 0BC4 ENABLED
GRAF 0BC5 ENABLED , GRAF 0BC6 ENABLED , GRAF 0BC7 ENABLED , GRAF 0BC8 ENABLED
```

The responses for offline and boxed displays are in the same format.

Responses from Real Communication Lines:

If a real communication line is dedicated to a user, you see something like this:

```
LINE 00F0 ATTACHED TO OLIVER 0010
```

This response means the communication line with the real device number 00F0 is attached to OLIVER's virtual machine as the virtual device 0010. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

If there are communication lines that have users logged on at them or dialed through them, you see something like this:

```
LINE 00FA LOGON AS DAVID 0009
LINE 00FB DIALED TO PVM 00F9
```

This response tells you the user DAVID is logged on through the communication line at 00FA. The number 0009 is the virtual device number of the virtual console where DAVID is logged on. The second line tells you the terminal connected to communications line 00FB is dialed to PVM's virtual device number 00F9.

If there are free communication lines, you see something like this:

```
LINE 00C1 FREE , LINE 00C2 FREE , LINE 00C3 FREE , LINE 00C4 FREE
LINE 00C5 FREE , LINE 00C6 FREE , LINE 00C7 FREE , LINE 00C8 FREE
LINE 00E1 FREE , LINE 00E2 FREE , LINE 00E3 FREE , LINE 00E4 FREE
LINE 00E5 FREE , LINE 00E6 FREE , LINE 00C0 FREE
```

Responses for boxed and offline communication lines are in the same format.

Responses from Open Systems Adapter Devices:

If there are Open Systems Adapter devices attached to the system at the addresses specified, you see something like this:

```
q 480-486
OSA 0480 ATTACHED TO PVM 0480
OSA 0481 ATTACHED TO PVM 0481
OSA 0482 ATTACHED TO RSCS 0482
OSA 0483 ATTACHED TO RSCS 0483
OSA 0484 ATTACHED TO PVM 0485
OSA 0485 ATTACHED TO PVM 0484
OSA 0486 ATTACHED TO VTAM 0486
Ready;
```

The first number is the real device number of the Open Systems Adapter devices. The ATTACHED operand might also read BOX/ATTC, meaning that although the device is attached and dedicated to a user, it is unavailable. PVM, RSCS, TCP/IP, and VMNET are user IDs to which the OSA devices are dedicated. The last number is the virtual device number which each virtual machine uses to refer to an OSA device.

If there are free OSA devices attached to the system at the specified addresses, you see something like this:

```
q 4a1-4bf free
OSA 04A1 FREE , OSA 04A2 FREE , OSA 04A3 FREE , OSA 04A4 FREE
OSA 04A4 FREE , OSA 04A5 FREE , OSA 04A9 FREE , OSA 04AA FREE
OSA 04AB FREE , OSA 04BC FREE , OSA 04BD FREE , OSA 04BE FREE
OSA 04BF FREE
```

This response shows the real device number for each free OSA device.

The responses for offline and boxed devices are in the same format.

Responses from Real Dynamic Switching Devices:

If there is a dynamic switching device dedicated to a user, you see something like this:

```
SWCH 0CD5 ATTACHED TO EMILY 0012
```

This means the switching device at the real device number 0CD5 is attached to EMILY's virtual machine as the virtual device number 0012. ATTACHED may be replaced by BOX/ATTC, which indicates that the dedicated switching device is not available (usually because of a hardware condition).

If there are free switching devices attached to the system, you see something like this:

```
SWCH 0387 FREE , SWCH 03A1 FREE , SWCH 03A2 FREE , SWCH 03A3 FREE
SWCH 03A4 FREE , SWCH 03A5 FREE , SWCH 03A9 FREE , SWCH 03AA FREE
```

This response shows the real device number for each free switching device. Responses for boxed and offline switching devices are in the same format.

Responses from Real Magnetic Tape Drives:

If there is a tape drive dedicated to a user, you see something like this:

```
TAPE 0102 ATTACHED TO RALPH 0236 R/W
```

This means that the magnetic tape drive at the address 0102 is attached to RALPH's virtual machine as the virtual device number 0236. R/W means that the user RALPH has read-write access to the tape drive. R/W might also read R/O, which means the user has read-only access.

If the tape drive is currently being used by the TRSAVE command, you see something like this:

```
TAPE 0242 ASSIGNED SYSTEM TRSAVE
```

If a tape that is not dedicated to a user is not available (boxed), you see something like this:

```
TAPE 0244 BOXED
```

If a tape drive has been attached MULTIUSER to one or more users, you see something like this:

```
TAPE 2100 ATTACHED TO USER1 0181 R/W NOASSIGN MULTIUSER
TAPE 2100 ATTACHED TO USER2 0181 R/W NOASSIGN MULTIUSER
```

This means that the tape drive at real address 2100 is attached to USER1 and USER2 at virtual device address 0181. MULTIUSER indicates that the tape drive is shared by both users, and NOASSIGN states that the users must manage the assignment of the tape drive by using assign and unassign I/O.

Responses for free or offline tape drives will be in the same format.

Responses from Real Unit Record Devices:

If a unit record device is dedicated to a user, you see something like this:

```
RDR 000C ATTACHED TO JAMES 02F0
```

This response means that the real card reader at the device number 000C is dedicated to JAMES's virtual machine at the virtual device number 02F0. RDR might be replaced by PRT or PUN, depending on the kind of unit record device. ATTACHED might be replaced by BOX/ATTC, which indicates that the device is attached to the virtual machine but is unavailable (usually because of a hardware problem).

*Responses from Active Readers:*

If a reader is assigned to the system for spooling activity and it is currently active with a spool file, you see a message like this:

```
RDR 000C STARTED SYSTEM
RDR 000C READING PERCY FILE 0021
```

000C is the real device number of the card reader. STARTED indicates the device is available for spooling activity. SYSTEM means the device is available for input with user identification cards. STARTED might be replaced by any of the following:

- DRAINING, which means the device would be unavailable for spooling activity as soon as it finished processing the current file.
- DRAINED, which means the device was unavailable for spooling activity. You could enter the CP START command to restart it.
- INT REQ, which means the device is not ready and requires manual intervention before spool file processing could resume.

**Note:** If STARTED were replaced by any of the above options, except DRAINING, the second line of the response would not appear.

SYSTEM might be replaced by a user ID, which means the device was available for input without user identification cards. All input decks would be assigned to the user ID specified.

The second line of the response means the card reader is currently active with a spool file. In this example, the reader at the real device number 000C is reading a card deck and creating a file that will be the user PERCY's spool file 0021.

Responses from Active Printers:

If there is an impact printer in use as a system spooling device, you see a response like this:

```
PRT 0E0E STARTED      SYSTEM CLASS 1          SECLABEL
PRT 0E0E FORM STANDARD MANUAL  SEP   N03800 DEFFCB  LIMIT NONE
PRT 0E0E FOLD  IMAGE IMAG4248 CHARS      FCB FCB8      LPP 068
PRT 0E0E DEST OFF      DEST - NONE - DEST - NONE - DEST - NONE -
PRT 0E0E NOAFP
```

This response tells you the following about the impact printer at the real device number 0E0E. The first line tells you the printer will:

- Be available for spooling activity
- Process files for any user ID
- Select only class 1 spool files for processing.

The second line tells you it will:

- Select files requesting the STANDARD form
- Wait for the operator to schedule forms output
- Generate a separator page for output files
- Not accept spool files that contain 3800 load CCWs
- Use the default forms control buffer that was loaded into it
- Process spool files regardless of their length.

The third line tells you it will:

- Translate (fold) lower case characters into upper case
- Use the system image library file IMAG4248
- Not use a character set
- Use the forms control buffer FCB8.
- Print separator pages with 68 lines per page.

The fourth line tells you the printer will not use destination values as a criterion for selecting files.

The fifth line tells you the printer will not process files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB). For more information about each line of the response, see the tables below.

If there is a 3800 printer in use as a system spooling device and that printer is currently active with a spool file, you see a response like this:

```
PRT 0F10 STARTED SYSTEM CLASS * SECLABEL
PRT 0F10 FORM STANDARD AUTO SEP ANY3800 LIMIT NONE
PRT 0F10 FLASH 0987 IMAGE IMAG3800 CHARS GF10 FCB 6 HOLD LPP 060
PRT 0F10 DEST OFF DEST - NONE - DEST - NONE - DEST - NONE -
PRT 0F10 AFP MARK
PRT 0F10 PRINTING DAVID FILE 0002 RECS 130K COPY 005 SEQ 001
```

This response tells you the following about the printer at the real device number 0F10. The first line means the printer will:

- Be available for spooling activity
- Process files for any user ID
- Select spool files regardless of class.

The second line tells you the printer will:

- Select files requesting the STANDARD form
- Automatically schedule forms output
- Generate a separator page for output files
- Accept spool files that contain 3800 load CCWs
- Process spool files regardless of their length.

The third line tells you the printer will:

- Use the forms overlay negative called 0987
- Use the system image library file IMAG3800
- Use the character set GF10
- Print six lines per inch
- Put spool files causing 3800 load checks into system HOLD status.
- Print separator pages with 60 lines per page.

The fourth line tells you that the printer will not use destination values as a criterion for selecting files.

The fifth line tells you that the printer will:

- Process files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB).
- Mark separator trailer pages with separator bars.

The last line tells you:

- The printer is active with the spool file 0002, owned by the user DAVID.
- The number of records remaining for output in the file is 130K. K is the number of records in thousands rounded to the nearest 1000. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million.
- There are five copies remaining for output.
- The sequence for the file on the printer is 001.

The parts of the first line of the response for impact printers and 3800 printers might also be as follows:

Category	Possible Values	Meaning
Status	STARTED	Device is available for spooling activity.
	DRAINING	Device is processing a file, but will be drained after the file is finished.
	DRAINED	Device is not available for spooling activity. Issue the CP START command to activate the device.
	INT REQ	Device is not available, and needs manual intervention before it can continue to process spool files.
	SETUP REQ	Printer is ready to print an alignment page so that the operator can align the forms properly on an impact printer. The operator must press the STOP and START buttons to print another alignment page, or enter the START command to begin printing the entire file.
	MOUNT REQ	Printer is waiting for the operator to mount a new form.
User ID	SYSTEM	Device processes files for any user ID.
	<i>userid</i>	Device processes files only for the specified user ID.
Spool file class	CLASS <i>cl</i>	Device services spool files of the specified classes, in the specified order. An output device can service as many as 8 output classes.

The parts of the second line of the response can be:

Category	Possible Values	Meaning
Form	FORM <i>form</i>	Device services the specified 1- to 8-character operator form number.

Category	Possible Values	Meaning
Scheduling	MANUAL	Operator schedules forms output. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is printed, the printer stops.
	AUTO	Operator wants CP to schedule forms output. The form that is currently active is processed first. When all files with the current form have been printed, CP selects a new form for the device.
	SETUP	<p>Operator wants CP to print a page for forms alignment verification. (Only valid for impact printers.) The operator receives a message to set up the form. After setting up the form, the operator must press the STOP button, then the START button. An alignment page prints with letters replaced by X's and numbers replaced by 9's on the first page of the file.</p> <p>Each time the operator presses STOP and START after a full alignment page is printed, another alignment page prints. After aligning the forms, the operator enters the START command for the device. The printer then returns to its previous mode (AUTO or MANUAL), and the file prints in its entirety.</p> <p><b>Note:</b> If neither MANUAL, AUTO, nor SETUP is specified and the printer is currently in SETUP mode, it returns to the previous mode, AUTO or MANUAL. The default forms output is MANUAL.</p>
File separator	SEP	File separator prints between output files.
	NOSEP	File separator does not print between output files.
3800 load CCWs	NO3800	Printer does not print any spool files containing 3800 load CCWs. This option ensures a fast output queue, because load CCWs issued to 3800 printers can take a lot of time.
	BEG3800	Only spool files without 3800 load CCWs, or spool files with 3800 load CCWs only at the beginning of the file, print on this device. This allows the user an initial setup with which the entire file can be printed.
	ANY3800	Output device will process spool files regardless of whether they contain 3800 load CCWs. This option may cause a slower output queue, based on the number of 3800 load CCWs issued by the spool files.
Default FCB	DEFFCB	Printer uses the default FCB with which it was loaded. CP does not send imbedded LOAD FCB CCWs to the printer. The LOADBUF command specifies the default FCB.
	FILEFCB	<p>CP sends FCBs to the real printer. File FCBs are LOAD FCB CCWs imbedded in the spool file. This is the default when you start the printer after a cold start.</p> <p><b>Note:</b> DEFFCB and FILEFCB only appear for non-1403 impact printers.</p>



Category	Possible Values	Meaning
Length limit	LIMIT NONE	CP selects files for printing regardless of the number of records.
	LIMIT <i>nnnnnnnnnn</i>	CP only selects spool files for printing that are less than or equal to the specified number of records. The spool file record number is the number of copies multiplied by the size of the file.

The parts of the third line of the response can be:

Category	Possible Values	Meaning
Folding into upper case	FOLD	CP translates (folds) lower case characters into upper case.
	NOFOLD	CP does not translate lower case characters into upper case. <b>Note:</b> FOLD and NOFOLD only appear for impact printers.
Forms overlay negative	FLASH <i>name</i>	Printer uses the specified forms overlay negative, which is from 1 to 4 characters long. <b>Note:</b> FLASH only appears for 3800 printers.
Image library	IMAGE <i>imagelib</i>	Printer uses the specified system image library file, which contains the load images (character sets, FCBs, graphic character modifications, or copy modifications) used for printing files and separator pages.
Character set	CHARS <i>chars</i>	Printer uses the specified character set, which is from 1 to 4 characters long, when printing the separator page.
FCB	FCB <i>name</i>	For impact printers: printer uses the specified FCB when printing the file or the separator page. For 3800 printers: the FCB is the number of lines per inch. <b>Note:</b> FCB does not appear for 1403-type printers.
3800 load checks	HOLD	Printer puts spool files causing 3800 load checks in system HOLD status.
	PURGE	Printer deletes spool files causing 3800 load checks. <b>Note:</b> HOLD and PURGE only appear for 3800 printers.
Lines per page	LPP	Defines the number of lines for file separator pages on the printer.

The parts of the next-to-last line of this response can be:

Category	Possible Values	Meaning
Destination	DEST	Printer services the specified destination values. The printer can service up to 4 destination values. The first value for the device is always OFF if it was not specifically set by either a START command, or at initialization with the RDEVICE statement in the system configuration file. If it was set, the specified destination is shown. If the remaining destination values were not specified, the values are null and are shown as – NONE –. If they were specified, the specified value is shown.
AFP characteristics	AFP	Printer processes files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB).
	NOAFP	Printer does not process files with AFP characteristics.
Mark separator trailer pages	MARK	Printer marks 3800 trailer pages with separator bars.
	NOMARK	Printer does not mark 3800 trailer pages with separator bars.

**Note:** MARK and NOMARK only appear for 3800 printers.

*Responses from Active Card Punches*

If a punch is attached to a user and is currently active with a spool file, you see a message like:

```
q 1f30
PUN 1F30 STARTED SYSTEM CLASS 1
PUN 1F30 FORM STANDARD AUTO NOSEP LIMIT NONE
PUN 1F30 DEST OFF DEST - NONE - DEST - NONE - DEST - NONE -
PUN 1F30 PUNCHING JAMES FILE 0042 RECS 012K COPY 002 SEQ 001
Ready;
```

This response tells you the following about the card punch at the real device number 1F30. The first line means that the punch will:

- Be available for spooling activity
- Process files for any user ID
- Select class 1 spool files.

The second line tells you that the punch will:

- Select files requesting STANDARD form
- Automatically schedule forms output
- Not generate a separator page for output files.

The third line tells you that the punch will not use destination values as a criterion for selecting files.

The parts of the first line of the response can also be:

Category	Possible Values	Meaning
Status	STARTED	Device is available for spooling activity.
	DRAINING	Device is processing a file, but will be drained when the file is finished.
	DRAINED	Device is not available for spooling activity. Issue the CP START command to activate the device.
	INT REQ	Device is not available; it needs manual intervention before it can continue to process spool files.
	SETUP REQ	Punch is ready to generate an alignment page to let the operator align the forms properly on an impact printer. The operator must press the STOP and START buttons to print another alignment page, or enter the START command to begin printing the entire file.
	MOUNT REQ	Punch is waiting for the operator to mount a new form.
User ID	SYSTEM	Device processes files for any user ID.
	<i>userid</i>	Device processes files only for the specified user ID.
Spool file class	CLASS <i>cl</i>	Device services spool files of the specified classes, in the specified order. An output device can service as many as 8 output classes.

The options for the second line of the response can be:

Category	Possible Values	Meaning
Form	FORM <i>form</i>	Device services the specified 1- to 8-character operator form number.
Scheduling	MANUAL	Operator schedules forms output. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is punched, the punch stops.
	AUTO	Operator wants CP to schedule forms output. The form that is currently active is processed first. After punching all files with the current form, CP selects a new form for the device.
File separator	SEP	File separator prints between output files.
	NOSEP	File separator does not print between output files.
Length limit	LIMIT NONE	CP selects files for punching regardless of the number of records.
	LIMIT <i>nnnnnnnnnn</i>	CP only selects spool files for punching that are less than or equal to the specified number of records. The spool file record number is the number of copies multiplied by the size of the file.

The options for the third line of the response can be:

Category	Possible Values	Meaning
Destination value	DEST	Printer services the specified destination values. The printer can service up to 4 destination values. The first value for the device is always OFF if it was not specifically set by either a START command, or at initialization with the RDEVICE statement in the system configuration file. If it was set, the specified destination is shown. If the remaining destination values were not specified, the values are null and are shown as – NONE –. If they were specified, the specified value is shown.

The last line of this response only appears if there is a file active on the card punch. The line in the example above tells you the punch at the real device number 1F30 is punching the spool file 0042, owned by the user JAMES. The number of records remaining for output in the file is 12K. K is the number of records in thousands rounded to the nearest 1000. If the number is greater than 999499, the number is shown as *nnnM*. M represents the number of lines in millions rounded to the nearest million. There are two copies remaining for output. The sequence for the file on the punch is 001.

Responses from Boxed Unit Record Devices:

If a unit record that is not dedicated to a user is unavailable, you see a message like the following:

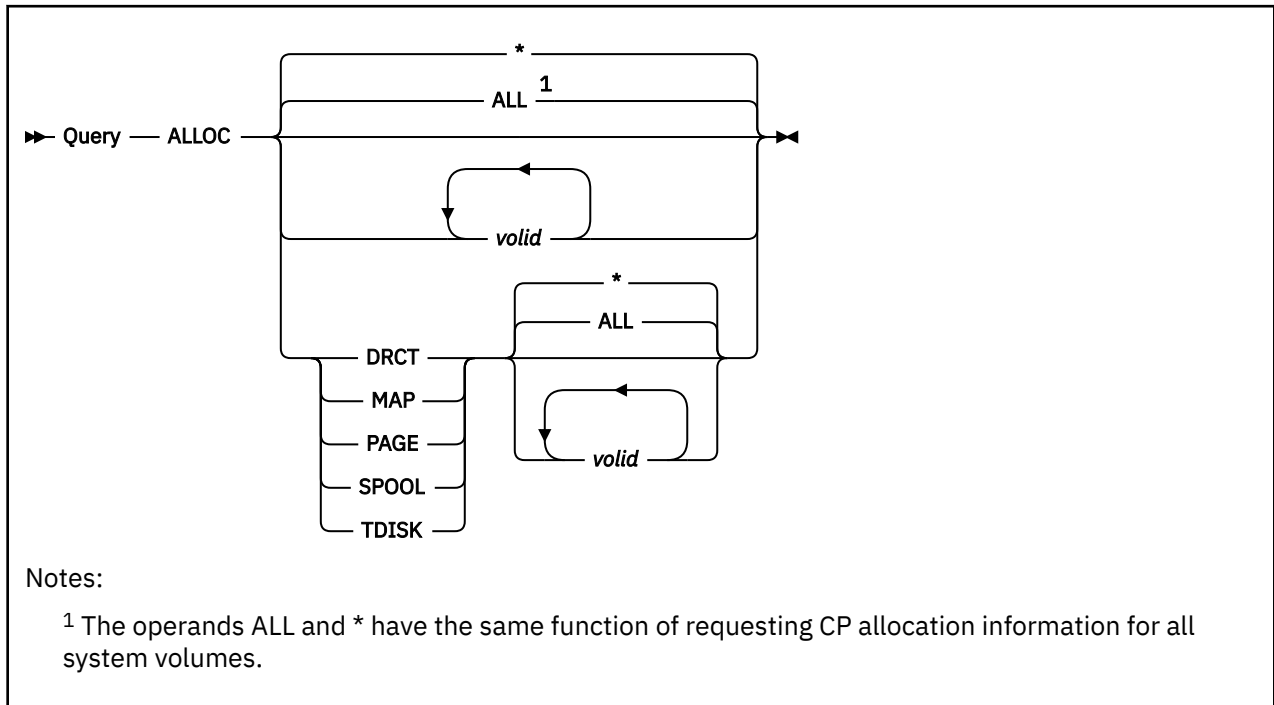
```
RDR 000F BOXED
```

000F is the real device number of the card reader that is unavailable. RDR might be replaced by PRT or PUN.

### Messages

- HCP003E Invalid option - *option*
- HCP020E Userid missing or invalid.
- HCP263E Too many operands were supplied for this command.

## QUERY ALLOC



### Authorization

Privilege Class: D

### Purpose

Use QUERY ALLOC to display the number of cylinders or pages that are allocated, in use, and available for DASD volumes attached to the system.

### Operands

#### ALL or \*

tells CP you want allocation information for all system volumes. This is the default.

#### valid

specifies one or more volume identifiers (volume IDs) for which you want information displayed. You can specify specific or generic volume IDs. To specify a generic volume ID, use the wildcard character (\*) as the last character of a string. CP compares the string with the beginning of all volume IDs on the system and gives you information about all volume IDs whose first few characters match the string.

#### DRCT

specifies that you want specific detailed information, including extent start and end cylinders or pages, and cylinder or page allocation, displayed for the DRCT extents on the specified volumes.

#### MAP

specifies that you want specific detailed information, including extent start and end cylinders or pages, and cylinder or page allocation, displayed for all CP use extents on the specified volumes.

#### PAGE

specifies that you want specific detailed information, including extent start and end cylinders or pages, and page allocation, displayed for the PAGE extents on the specified volumes.

### SPOOL

specifies that you want specific detailed information, including extent start and end cylinders or pages, and page allocation, displayed for the SPOOL extents on the specified volumes.

### TDISK

specifies that you want specific detailed information, including extent start and end cylinders or pages, and cylinder or page allocation, displayed for the TDISK extents on the specified volumes.

## Usage Notes

1. For options PAGE, SPOOL, TDISK, and DRCT, if *valid1* is not supplied, information is displayed for all CP-owned volumes that have space of that type allocated on them. For the MAP option, if *valid1* is not supplied, information is displayed for all CP use extents on all CP-owned volumes that have any CP use extents.
2. If QUERY ALLOC PAGE, QUERY ALLOC SPOOL, QUERY ALLOC TDISK, QUERY ALLOC DRCT, or QUERY ALLOC MAP commands are issued through DIAGNOSE code X'08' and the user requests the response returned in a buffer, complete volume information (*valid* and *rdev*) will be displayed for each extent displayed on the volume. This allows easier parsing of the command output by application execs or programs.
3. *Valid*s PAGE, SPOOL, TDISK, DRCT, MAP, and EXEC should be treated as restricted and should not be used for CP-owned volumes. If any of these *valid*s are used, the corresponding new option will not be available. For example QUERY ALLOC PAGE will default to the QUERY ALLOC *valid* syntax if *valid* PAGE is defined to CP.
4. Pattern wildcard characters (\*) are allowed only as the last character of the *valid*.
5. If space on a CP-owned volume is re-allocated while the volume is still attached to the system and a subsequent system IPL or the commands DETACH *rdev* FROM SYSTEM and ATTACH *rdev* TO SYSTEM do not occur, QUERY ALLOC may report incorrect information for that volume. This will occur until the system is IPLed or the device is DETACHED and re-ATTACHED to the system.
6. The *chnprg* field of the Query Alloc response in combination with other device/control unit hardware information can be used to determine if performance problems may be occurring. Severe performance degradation may occur on a device to which CP issues CKD channel programs (because of the presence of filler records) if the device is attached to an ECKD-capable control unit by way of an ESCON or non-synchronous interface. The problem can be corrected by formatting the DASD using the NO FILLER option of ICKDSF.

## Examples

Examples of QUERY ALLOC commands:

```
QUERY ALLOC
```

displays cylinder or page information about all system volumes.

```
QUERY ALLOC ALL
```

displays cylinder or page information about all system volumes.

```
QUERY ALLOC VOL001
```

displays cylinder or page information about a single system volume whose *valid* is VOL001.

```
QUERY ALLOC VOL001 DASD99
```

displays cylinder or page information about two system volumes whose *valid*s are VOL001 and DASD99.

```
QUERY ALLOC VOL*
```

displays cylinder or page information about all system volumes whose *valids* begin with the three characters VOL.

```
QUERY ALLOC VOL* DASD*
```

displays cylinder or page information about all system volumes whose *valids* begin with the three characters VOL or the four characters DASD.

```
QUERY ALLOC PAGE VOL001
```

displays detailed PAGE extent information for a single system volume whose *valid* is VOL001.

```
QUERY ALLOC PAGE VOL001 DASD99
```

displays detailed PAGE extent information for two system volumes whose *valids* are VOL001 and DASD99.

## Responses

Response 1:

```
DASD rdev valid type format{-chnprg} (UNITS IN units)
      TDISK TOTAL=nnnnnnnnnn INUSE=nnnnnnnnnn AVAIL=nnnnnnnnnn
      PAGE  TOTAL=nnnnnnnnnn INUSE=nnnnnnnnnn AVAIL=nnnnnnnnnn
      SPOOL TOTAL=nnnnnnnnnn INUSE=nnnnnnnnnn AVAIL=nnnnnnnnnn <,DR>
      DRCT  TOTAL=nnnnnnnnnn INUSE=nnnnnnnnnn AVAIL=nnnnnnnnnn <,<ACTIVE>
<IPL NUCLEUS ACTIVE ON VOLUME validn>
```

is displayed for each CP-owned volume.

### **rdev**

is the real device number of the CP-owned volume.

### **valid**

is the volume identification of the CP-owned volume.

### **type**

shows the DASD device type (for example, 3380).

### **format**

identifies the type of device (CKD or FBA).

### **<-chnprg>**

identifies the type of channel program that CP issues to the device (CKD or ECKD). This field is only displayed for CKD devices. The type of channel programs issued to a CKD device depends on the device type, the control unit to which it is attached, and if the device was formatted with filler records. CP will issue CKD channel programs to a 3380 attached to an ECKD capable control unit if it was formatted using the FILLER option of ICKDSF (filler records exist). CP will issue ECKD channel programs to the same device/control unit combination if it was formatted with the NO FILLER option of ICKDSF. For best performance, the type of channel programs that CP issues to the device should match the interface of the control unit to which it is attached. See the usage notes section for more details.

### **units**

identifies the units of *nnnnnnnnnn* in the response for this volume. *units* is set to CYLINDERS for CKD devices and PAGES for FBA devices.

### **nnnnnnnnnn**

shows the number of cylinders or pages allocated in each category. For PAGE and SPOOL, when *units* is set to CYLINDERS, a cylinder is considered "in use" when at least one page in it is in use. Calculating a percentage used -  $(INUSE * 100) / (INUSE + AVAIL)$  - will therefore usually yield a larger and less accurate result than the percentage used shown in the following responses.

### **<,<ACTIVE>**

appears on the DRCT response line if that volume contains the currently active user directory.

## QUERY ALLOC

### <DR>

appears on the TDISK, PAGE, or SPOOL response line if CP is currently not allocating space of that type on that volume.

### <IPL NUCLEUS ACTIVE ON VOLUME *valid*>

appears as the final response line if the volume containing the active IPL nucleus was one of those for which cylinder or page allocation information was displayed.

Response 2:

VOLID	RDEV	EXTENT START	EXTENT END	TOTAL PAGES	PAGES IN USE	HIGH PAGE	% USED	
<i>valid</i>	<i>rdev</i>	nnnnnnnnnn	nnnnnnnnnn	pppppp	pppppp	pppppp	ppp%	<DR>
		nnnnnnnnnn	nnnnnnnnnn	pppppp	pppppp	pppppp	ppp%	<DR>
		nnnnnnnnnn	nnnnnnnnnn	pppppp	pppppp	pppppp	ppp%	<DR>
<i>valid</i>	<i>rdev</i>	nnnnnnnnnn	nnnnnnnnnn	pppppp	pppppp	pppppp	ppp%	DUMP <DR>
		nnnnnnnnnn	nnnnnnnnnn	pppppp	pppppp	pppppp	ppp%	DUMP <DR>
<i>valid</i>	<i>rdev</i>	-	-	-	-	-	-	SHARED
<i>valid</i>	<i>rdev</i>	-	-	0	0	0	0%	NOT FOUND
SUMMARY				pppppp	pppppp		ppp%	
usable				pppppp	pppppp		ppp%	
DRAINING				pppppp	pppppp		ppp%	

is the response for QUERY ALLOC PAGE and QUERY ALLOC SPOOL.

### *valid*

is the volume identification of the CP-owned volume.

### *rdev*

is the real device number of the CP-owned volume.

### *nnnnnn*

is the starting or ending cylinder or page number of the extent. For CKD volumes, this number is in cylinders. For FBA volumes, this number is in pages. The abbreviations K, M, G, T and P are used to indicate Kilo, Mega, Giga, Tera and Peta, for example, nnnnnK.

### *pppppp*

is the total number of pages in the extent, the number of pages in use in the extent, or the highest page number in use in the extent. The abbreviations K, M, G, T and P are used to indicate Kilo, Mega, Giga, Tera and Peta, for example, pppppK.

### *ppp*

is the percentage of pages in use in the extent.

### DUMP

indicates that this SPOOL extent is reserved for DUMP use only.

### SHARED

indicates the volume is owned by another system in the same SSI cluster. This system cannot allocate or release space on the volume, but it can read spool data from areas previously allocated and written to by the owning system. No allocation data can be displayed for this volume.

### Notes:

1. The SHARED response line will be displayed only for QUERY ALLOC SPOOL or QUERY ALLOC MAP, unless that volume was specifically requested.
2. A zero (0) will be displayed in the TOTAL, IN USE, HIGH, and % USED fields of this line.

### <DR>

indicates the extent is draining. If DR is indicated, CP is currently not allocating space of that type on that volume. Draining applies only to TDISK, PAGE, and SPOOL space.

### NOT FOUND

indicates that no CP-owned extents of the requested allocation type (PAGE or SPOOL) were found on the specified *valid*. The *valid* is the *valid* operand specified on the QUERY ALLOC command that was issued. It can, therefore, be a wildcard *valid* specifier, in which case it indicates that no CP-owned



extents of the requested type were found on any volume with a *valid* matching the wildcard *valid* specifier. In this case, the *rdev* is displayed as *\**.

**Notes:**

1. Leading zeros are not displayed for *nnnnnn*, *pppppp* and *ppp*.
2. *Valid* and *rdev* are only displayed for the first extent on the volume, unless the command was issued using DIAGNOSE code X'08' and the response was requested in the user's buffer. In this case *valid* and *rdev* are displayed on every response line.
3. The summary response lines (including the footer line) are displayed only when information for all system volumes is displayed (the default ALL or \* was specified as the only operand) and some space of the requested type was found. The SUMMARY line gives system-wide total, in-use, and percent in-use values, including both usable and draining space. The usable line gives system-wide total, in-use, and percent in-use values, excluding any space not generally usable for the specified type (either draining for the specified type, or for the type SPOOL, reserved for dump use only). The DRAINING line is displayed only if draining space of the specified type (that would otherwise be generally usable) was found. The DRAINING line gives system-wide total, in-use, and percent in-use values for all draining volumes that are draining for the specified type (and for type SPOOL, not reserved for dump use only).

Response 3:

VOLID	RDEV	EXTENT START	EXTENT END	TOTAL	IN USE	HIGH	% USED	
<i>valid</i>	<i>rdev</i>	<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>ppp%</i>	<DR>
		<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>ppp%</i>	<DR>
<i>valid</i>	<i>rdev</i>	<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>ppp%</i>	ACTIVE
<i>valid</i>	<i>rdev</i>	-	-	-	-	-	-	SHARED
<i>valid</i>	<i>rdev</i>	-	-	0	0	0	0%	NOT FOUND
SUMMARY				< <i>nnnnnn</i>	<i>nnnnnn</i>		<i>ppp%</i>	CKD>
usable				< <i>nnnnnn</i>	<i>nnnnnn</i>		<i>ppp%</i>	FBA>
DRAINING				< <i>nnnnnn</i>	<i>nnnnnn</i>		<i>ppp%</i>	CKD>
				< <i>nnnnnn</i>	<i>nnnnnn</i>		<i>ppp%</i>	FBA>

is the response for QUERY ALLOC DRCT and QUERY ALLOC TDISK.

***valid***

is the volume identification of the CP-owned volume.

***rdev***

is the real device number of the CP-owned volume.

***nnnnnn***

is the starting or ending cylinder or page number of the extent, the total number of cylinders or pages in the extent, the number of cylinders or pages in use in the extent, or the highest cylinder or page number in use in the extent. For CKD volumes, this number is in cylinders. For FBA volumes, this number is in pages. The abbreviations K, M, G, T and P are used to indicate Kilo, Mega, Giga, Tera and Peta, for example, *nnnnnK*.

***ppp***

is the percentage of cylinders or pages in use in the extent.

**ACTIVE**

indicates this DRCT extent contains the currently active user directory.

**SHARED**

indicates the volume is owned by another system in the same SSI cluster. This system cannot allocate or release space on the volume, but can read spool data from areas previously allocated and written to by the owning system. No allocation data can be displayed for this volume.

**Notes:**

## QUERY ALLOC

1. The SHARED response line will be displayed only for QUERY ALLOC SPOOL or QUERY ALLOC MAP, unless that volume was specifically requested.
2. A zero (0) will be displayed in the TOTAL, IN USE, HIGH, and % USED fields of this line.

### <DR>

indicates the extent is draining. If DR is indicated, CP is currently not allocating space of that type on that volume. Draining applies only to TDISK, PAGE, and SPOOL space.

### NOT FOUND

indicates no CP-owned extents of the requested allocation type (TDISK or DRCT) were found on the specified *valid*. The *valid* is the *valid* operand specified on the QUERY ALLOC command that was issued. It can, therefore, be a wildcard *valid* specifier, in which case it indicates that no CP-owned extents of the requested type were found on any volume with a *valid* matching the wildcard *valid* specifier. In this case, the *rdev* is displayed as \*.

### Notes:

1. Leading zeros are not displayed for *nnnnnn* and *ppp*.
2. *Valid* and *rdev* are only displayed for the first extent on the volume, unless the command was issued using DIAGNOSE code X'08' and the response was requested in the user's buffer. In this case, *valid* and *rdev* are displayed on every response line.
3. The summary response lines (including the footer line) are displayed only when information for all system volumes is displayed (the default ALL or \* was specified as the only operand) and some space of the requested type was found. The summary information gives system-wide total, in-use, and percent in-use values. The summary information is divided into three sections: SUMMARY, usable, and DRAINING. If DRCT was specified, only the SUMMARY section is displayed. If TDISK was specified, the SUMMARY and usable sections are displayed. The DRAINING section is displayed only if draining TDISK space was found. If DRCT was specified, the SUMMARY section gives system-wide total, in-use, and percent in-use values.

If TDISK was specified, the SUMMARY section gives system-wide total, in-use, and percent in-use values (including both usable and draining TDISK space); the usable section gives system-wide total, in-use, and percent in-use values (excluding any space on volumes that are draining for TDISK); and the DRAINING section gives system-wide total, in-use, and percent in-use values for all draining volumes that are draining for TDISK.

The first line of each section is a summary of all CKD volumes, and the information is displayed in cylinders. The second line of each section is a summary of all FBA volumes and the information is displayed in pages. If no information was displayed for CKD volumes, then no CKD summary line is displayed; only the FBA summary line is displayed. If no information is displayed for FBA volumes, then no FBA summary line is displayed; only the CKD summary line is displayed.

Response 4:

VOLID	RDEV	EXTENT START	EXTENT END	TOTAL	IN USE	HIGH	% USED	ALLOCATION TYPE
<i>valid</i>	<i>rdev</i>	<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>pppppp</i>	<i>pppppp</i>	<i>pppppp</i>	<i>ppp%</i>	PAGE <DR>
		<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>ppp%</i>	DRCT ACTIVE
		<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>pppppp</i>	<i>pppppp</i>	<i>pppppp</i>	<i>ppp%</i>	SPOOL <DR>
		<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>ppp%</i>	TDISK <DR>
		<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>pppppp</i>	<i>pppppp</i>	<i>pppppp</i>	<i>ppp%</i>	SPOOL <DR>
<i>valid</i>	<i>rdev</i>	<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>pppppp</i>	<i>pppppp</i>	<i>pppppp</i>	<i>ppp%</i>	SPOOL DUMP <DR>
		<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>nnnnnn</i>	<i>ppp%</i>	DRCT
		<i>nnnnnnnnnn</i>	<i>nnnnnnnnnn</i>	<i>pppppp</i>	<i>pppppp</i>	<i>pppppp</i>	<i>ppp%</i>	SPOOL DUMP <DR>
<i>valid</i>	<i>rdev</i>	-	-	-	-	-	-	SHARED
<i>valid</i>	<i>rdev</i>	-	-	0	0	0	0%	NOT FOUND

is the response for QUERY ALLOC MAP.

### **valid**

is the volume identification of the CP-owned volume.

**rdev**

is the real device number of the CP-owned volume.

**nnnnnn**

is the starting or ending cylinder or page number of the extent, the total number of cylinders or pages in the extent, the number of cylinders or pages in use in the extent, or the highest cylinder or page number in use in the extent. For CKD volumes, this number is in cylinders. For FBA volumes, this number is in pages. The abbreviations K, M, G, T and P are used to indicate Kilo, Mega, Giga, Tera and Peta, for example, nnnnnK.

**pppppp**

is the total number of pages in the extent, the number of pages in use in the extent, or the highest page number in use in the extent. The abbreviations K, M, G, T and P are used to indicate Kilo, Mega, Giga, Tera and Peta, for example, pppppK.

**ppp**

is the percentage of pages (for all FBA response lines and for CKD PAGE and SPOOL response lines) or cylinders (for CKD DRCT and TDISK response lines) in use in the extent.

**PAGE**

indicates the extent is of CP-use allocation type PAGE.

**SPOOL**

indicates the extent is of CP-use allocation type SPOOL.

**DRCT**

indicates the extent is of CP-use allocation type DRCT.

**TDISK**

indicates the extent is of CP-use allocation type TDISK.

**DUMP**

indicates this SPOOL extent is reserved for DUMP use only.

**ACTIVE**

indicates this DRCT extent contains the currently active user directory.

**SHARED**

indicates the volume is owned by another system in the same SSI cluster. This system cannot allocate or release space on the volume, but it can read spool data from areas previously allocated and written to by the owning system. No allocation data can be displayed for this volume.

**Notes:**

1. The SHARED response line will be displayed only for QUERY ALLOC SPOOL or QUERY ALLOC MAP, unless that volume was specifically requested.
2. A zero (0) will be displayed in the TOTAL, IN USE, HIGH, and % USED fields of this line.

**<DR>**

indicates the extent is draining. If DR is indicated, CP is currently not allocating space of that type on that volume. Draining applies only to TDISK, PAGE, and SPOOL space.

**NOT FOUND**

indicates no CP-owned extents (allocation type PAGE, SPOOL, TDISK, or DRCT) were found on the specified *valid*. The *valid* is the *valid* operand specified on the QUERY ALLOC command that was issued. It can, therefore, be a wildcard *valid* specifier, in which case it indicates that no CP-owned extents of the requested type were found on any volume with a *valid* matching the wildcard *valid* specifier. In this case, the *rdev* is displayed as \*.

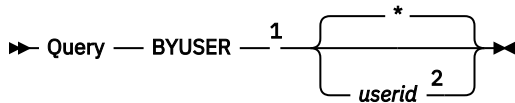
**Notes:**

1. Leading zeros are not displayed for *nnnnnn*, *pppppp* and *ppp*.
2. *Valid* and *rdev* are only displayed for the first extent on the volume, unless the command was issued using DIAGNOSE code X'08', and the response was requested in the user's buffer. In this case, *valid* and *rdev* are displayed on every response line.
3. Any volume extents not displayed by QUERY ALLOC MAP are allocated as PERM and PARM space.

**Messages**

- HCP013E Conflicting option - *option*
- HCP1002E Volume identifier *valid* does not exist.

## QUERY BYUSER



### Notes:

<sup>1</sup> If QUERY BYUSER is executed with the AT command, the *userid* operand is required.

<sup>2</sup> A non-class B or E user may specify only the user's own user ID.

## Authorization

Privilege Class: Any, B, E

## Purpose

Use the QUERY BYUSER command to display the BYUSER ID defined for a virtual machine. If you are not a class B or E user, you can display only the BYUSER ID for your own virtual machine. If you are a class B or E user, you can display the BYUSER ID for any virtual machine.

## Operands

\*

displays the BYUSER ID of the user issuing the command.

*userid*

displays the BYUSER ID of the specified virtual machine.

## Usage Notes

1. A BYUSER ID is defined only for a virtual machine that is logged on with the BY option of the LOGON command.
2. When a virtual machine is disconnected, its BYUSER ID remains unchanged.
3. A virtual machine's BYUSER ID is updated on a successful reconnect.

## Responses

Response 1:

```
The BYUSER for userid is byuserid
```

is returned when there is a BYUSER ID for the virtual machine specified in the query.

Response 2:

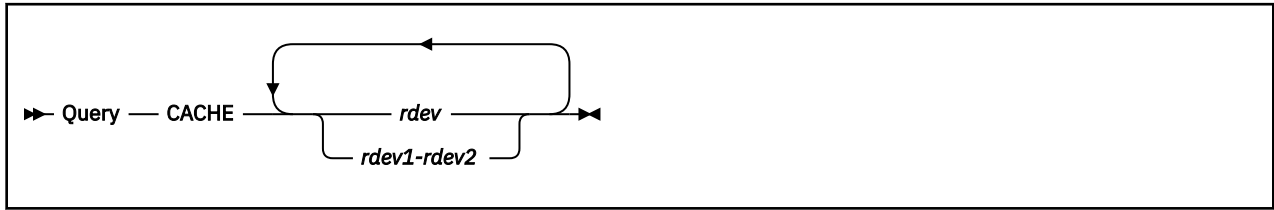
```
There is currently no BYUSER for userid
```

is returned when there is no BYUSER ID for the virtual machine specified in the query.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid *userid* - *userid*
- HCP045E *userid* not logged on

## QUERY CACHE



### Authorization

Privilege Class: B

### Purpose

Use QUERY CACHE to display caching status for all storage subsystems that support caching.

### Operands

*rdev*

*rdev1-rdev2*

is the real device number of one of the devices in the subsystem, a range of real device numbers, or any combination thereof.

### Usage Notes

1. When the cache is queried for a range of devices or a list of devices attached to an IBM DASD storage controller and the devices in the range or list are attached to the same storage controller, only one subsystem response is issued for each storage controller.

For example, to display the caching status for the cache associated with device 340, enter:

```
query cache 340
```

2. Caching for RAID (Redundant Array of Independent Disks) technology DASD such as an IBM DASD subsystem is always available. It cannot be modified by the user.
3. Error message HCP296E appears if the subsystem is not in the correct mode to accept the QUERY CACHE command.
4. This command may be issued to offline devices.

### Responses

Response 1:

```
rdev CACHE {available    for subsystem}
           {activation for subsystem in progress}
           {unavailable  for subsystem}

[- destage is in progress          ]
[- destage has failed. Data is pinned.]
[- subsystem error                 ]

cccccccK Bytes configured
aaaaaaaK Bytes available
oooooooK Bytes offline
pppppppK Bytes pinned
```

is the response for each IBM DASD subsystem.

If the cache is unavailable:

- `Destage is in progress` is included in the response if the storage control is still in the process of moving changed data from the cache to the backing storage device.
- If an error is encountered during the destaging process, `destage has failed`. Data is pinned. is displayed and the data that encountered the failure is held in the storage control cache.
- `Subsystem error` is included if an internal subsystem error occurred which caused termination of caching.

```
rdev CACHE {activated    for device          }
           {deactivated  for device          }
           {deactivation for device is pending.}
           Destage has failed.}
```

```
[rdev has pinned data]
```

is the response for each device specified in the command; if pinned data exists for the device. The variables `rdev`, `ccccccc`, and `aaaaaaaa` are the same as in Response 1.

**pppppppp**

is the size in kilobytes of subsystem storage that is not available to the storage director for allocation because of DASD exception conditions which prevent successful completion of a subsystem storage to DASD transfer.

**Note:** `ccccccc`, `aaaaaaaa`, `oooooooo`, and `pppppppp` are displayed in decimal.

Response 2:

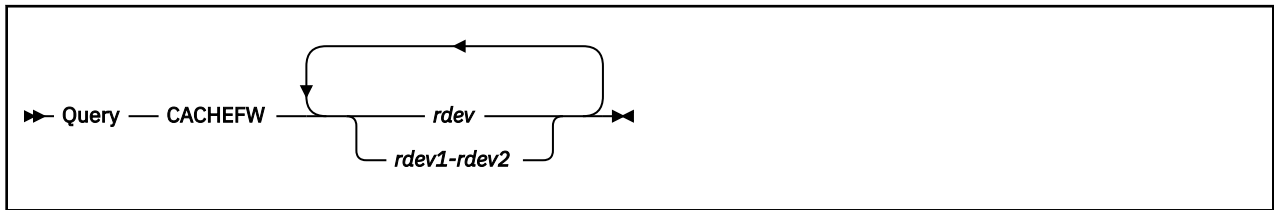
```
Bytes {available  }
      {unavailable} cannot be determined
      {offline   }
      {bound     }
      {pinned    }
```

is displayed if the storage control is unable to determine the subsystem storage capacity.

**Messages**

- HCP006E Invalid device type - `{rdev/vdev/ldev}`
- HCP009E Invalid range - `range`
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device `{rdev/vdev/ldev}` does not exist
- HCP296E Status is not as required - `rdev; text`
- HCP319E Error processing subsystem command - `cmd` for DASD `rdev`
- HCP332E Invalid control unit type - `rdev`
- HCP1120E The requested `[command|DETACH|GIVE]` for device `[ldev|rdev]` did not complete in the allotted time.
- HCP1150E DASD `{rdev/vdev}` is not a valid base exposure.
- HCP2600I DASD `rdev` subsystem storage status cannot be determined.
- HCP6867E Device `rdev` is an alias HyperParallel Access Volume.

## QUERY CACHEFW



### Authorization

Privilege Class: B

### Purpose

Use the QUERY CACHEFW command to display the cache fast write status for cache storage subsystems.

### Operands

*rdev*

*rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers of devices in the subsystems being queried.

### Usage Notes

1. The QUERY CACHEFW command is only valid for cache storage subsystems that support the cache fast write function. An error message is generated if the specified subsystem does not support this function.

### Responses

Response 1:

```
rdev Cache fast write function is {active. }
                                {inactive.}
                                {suspended.}
```

This response is repeated for each **unique subsystem** that is associated with the devices specified when you enter the QUERY CACHEFW command.

*rdev*

is the real device number of one of the specified devices in a subsystem being queried.

**active**

indicates that the cache fast write function is active for the subsystem.

**inactive**

indicates that the cache fast write function is inactive for the subsystem.

**suspended**

indicates that the cache fast write function is suspended for the subsystem. Suspension occurs when an authorized guest issues the Perform Subsystem Function channel command to suspend all fast write operations.

### Messages

- HCP006E Invalid device type - {rdev/vdev/ldev}



- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*

## QUERY CAPABILITY

► Query — CAPability ◄

### Authorization

Privilege Class: A, B, C, E

### Purpose

Use the QUERY CAPABILITY command to determine the capability values of the processors in the configuration. A processor's capability value indicates its capability relative to that of other CPU models. For more information on CPU capability, see [z/Architecture Principles of Operation](#).

### Responses

Response 1:

```
CAPABILITY: PRIMARY pppppppp SECONDARY ssssssss [NOMINAL nnnnnnnn]
[CAPACITY-ADJUSTMENT INDICATION cai CAPACITY-CHANGE REASON ccr]
[capacity_change_explanation]
```

#### ***pppppppp***

is the capability of a primary CPU in the configuration. See usage note “1” on page 787.

#### ***ssssssss***

is the capability of a secondary CPU in the configuration. See usage note “1” on page 787.

#### ***nnnnnnnn***

is the nominal CPU capability for the configuration. This value is displayed only if it is nonzero. See usage note “1” on page 787.

#### ***cai***

is the capacity-adjustment indication. This decimal value is in the range 1 - 100 and represents the aggregate position of model-dependent controls. When the value is zero, the response line is not displayed. The source of the value is Store System Information (STSI) 1.1.1.

#### ***ccr***

is the capacity-change reason, which explains why the capacity change occurred. This value is the *ccr* value returned by STSI 1.1.1. When the *cai* value is zero, this response line is not displayed.

#### ***capacity\_change\_explanation***

is an explanation of the *cai* and *ccr* values. When the *cai* value is zero, this response line is not displayed. The text of the response will be one of the following:

**RUNNING AT NOMINAL CAPACITY.**

**RUNNING WITH REDUCED CAPACITY DUE TO A MANUAL CONTROL SETTING.**

For example, static power savings mode.

**RUNNING WITH REDUCED CAPACITY DUE TO A MACHINE EXCEPTION CONDITION.**

For example, a cooling problem.

**RUNNING WITH REDUCED CAPACITY DUE TO A MACHINE NON-EXCEPTION CONDITION.**

For example, a firmware update.

**RUNNING WITH REDUCED CAPACITY DUE TO AN EXTERNAL EXCEPTION CONDITION.**

For example, ambient temperature exceeded the specified maximum.

**MACHINE CAPACITY INDICATORS ARE IN AN UNEXPECTED STATE.**

The *ccr* value is not recognized.

**MACHINE CAPACITY IS IN TRANSITION BETWEEN STATES.**

A change to one or more machine capacity-related values is expected to occur soon.

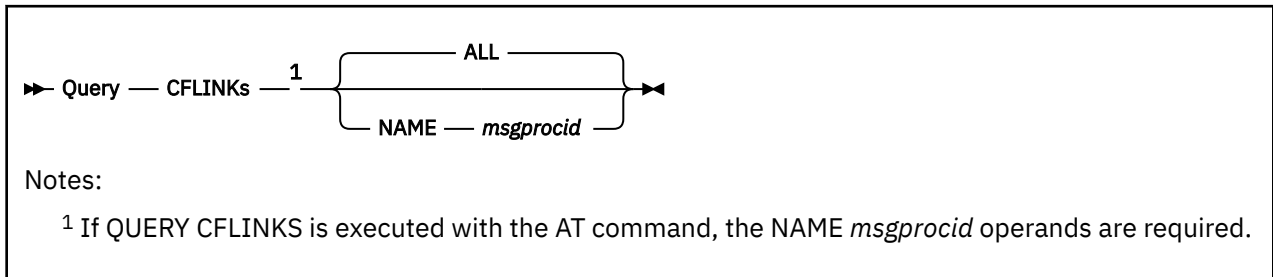
**Usage Notes**

1. The primary, secondary, and nominal capacity values returned in the response can be integers or decimal fractions (such as, 5.125678). In some cases the value might be too large or too small to display, in which case the term "OVERFLOW" or "UNDERFLO" will be displayed.

**Messages**

- HCP263E Too many operands were supplied for this command.

## QUERY CFLINKS



### Authorization

Privilege Class: A, B, G

### Purpose

Use QUERY CFLINKS to display information about links between coupling facilities.

### Operands

#### ALL

specifies that coupling facility to coupling facility link information should be displayed for all coupling facility virtual machines to which the issuer is coupled.

#### NAME *msgprocid*

indicates a specific coupling facility virtual machine that is to be queried.

### Usage Notes

1. If the caller is a class G user, then only coupling facility-to-coupling link information for message processors within the user's current I/O configuration is displayed. A class A or B user may display coupling facility link information for a particular message processors by using the NAME keyword.
2. QUERY CFLINK returns information about the coupling facility-to-coupling facility links that were created by the DEFINE CFLINK command or the SPECIAL MSGP directory statement for a coupling facility virtual machine. The QUERY MSGP command may be used to obtain information for coupling links between OPTION CFUSER and coupling facility virtual machines.

### Responses

The command first produces two lines of output describing the queried coupling facility virtual machine:

```
MESSAGE PROCESSOR=nn   USERID=userid   PEERS=pp
TYPE=SIMDEV  MODEL=001  MFG=IBM  PLANT=EN  SEQ=ND01
```

where:

#### *nn*

is a sequence number assigned to the coupling facility virtual machine by QUERY CFLINKS.

#### *userid*

is the user ID of the coupling facility virtual machine being queried.

#### *pp*

is either the number of sender peers or the number of receiver peers of this coupling facility, whichever is greater.

#### TYPE

is the six-character device type in the worldwide unique ID of the message processor's node descriptor.

**MODEL**

is the three-character model number in the worldwide unique ID of the message processor's node descriptor.

**MFG**

is the three-character code that identifies the manufacturer in the worldwide unique ID of the message processor's node descriptor.

**PLANT**

is the two-character code that identifies the plant of manufacture in the worldwide unique ID of the message processor's node descriptor.

**SEQ**

is the 12-character code that contains the serial number in the worldwide unique ID of the message processor's node descriptor.

Two lines of output are produced for each coupling facility virtual machine to which this coupling facility virtual machine is connected. The first appears as follows:

```
DEVICES=nnnn nnnn ...
```

These are the virtual devices in use as message devices to the target coupling facility virtual machine.

The second of these lines is displayed:

```
CFS=aa bb (PA=dd) CFR=ee ff (PA=hh) S_RLINK=userid
```

where:

**aa**

The sending side's first channel path ID.

**bb**

The sending side's second channel path ID (that might not appear).

**dd**

The sender's path available mask.

**ee**

The receiving side's first channel path ID.

**ff**

The receiving side's second channel path ID (that might not appear).

**hh**

The receiver's path available mask.

**userid**

The user ID of the receiving coupling facility virtual machine.

This pair of lines is repeated for each coupling facility virtual machine to which the queried coupling facility virtual machine is coupled.

**Examples**

CFUSER1 is coupled to coupling facility CFVM1. To find information about the coupling facility-to-coupling facility links for which CFVM1 initiated, CFUSER1 could enter the following:

```
query cflink name cfvm1
```

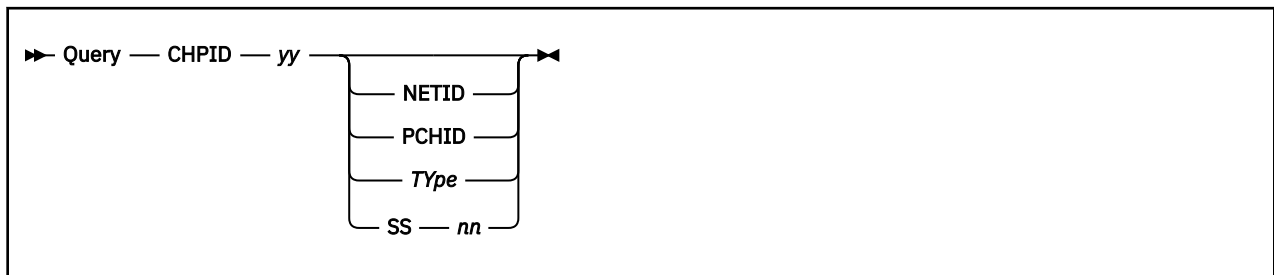
To find coupling facility-to-coupling facility link information for all coupling facility virtual machines to which CFUSER1 is coupled, CFUSER1 could enter the following:

```
query cflink
```

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP2800E *msgprocid* is not an authorized CF Service Machine.
- HCP2805E One or more message processors does not exist in your I/O configuration.
- HCP2806E You are not allowed to control message processors.
- HCP2820E You are not coupled to any CF service machines.

## QUERY CHPID



### Authorization

Privilege Class: B

### Purpose

Use QUERY CHPID to display the following:

- All devices in the active configuration with logical channel path *yy* installed
- Logical path status as it pertains to those devices (online, offline, or quiesced).

Use QUERY CHPID *yy* NETID to display the network identifier associated with CHPID *yy*.

Use QUERY CHPID *yy* PCHID to display the physical channel identifier associated with CHPID *yy*.

Use QUERY CHPID *yy* TYPE to display the channel path type of CHPID *yy*.

Use QUERY CHPID *yy* SS *nn* to display the following:

- All devices in the subchannel set *nn* with logical change path *yy* installed.
- Logical path status as it pertains to those devices (online, offline, or quiesced).

### Operands

#### *yy*

is the logical channel path whose device allocation information is requested. The variable *yy* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

#### NETID

requests the display of the network identifiers associated with CHPID *yy*.

#### PCHID

requests the display of the physical channel identifier associated with CHPID *yy*.

**Note:** This parameter is valid only when your machine is capable of logical channel-subsystem support.

#### TYpe

requests the display of the channel path type of CHPID *yy*.

#### SS *nn*

requests the display of devices in subchannel set *nn* with logical channel path *yy* installed. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'03'.

### Usage Notes

1. The QUERY CHPID command shows all devices associated with a specific logical channel path. For information about physical channel paths, use the QUERY CHPIDS command. (See [“QUERY CHPIDS”](#))

on page 796.) For information about logical and physical channel paths associated with a specific device, use the QUERY PATHS command. (See “QUERY PATHS” on page 1064.)

2. If a device has never been initialized, its path information is not available and, therefore, cannot be included in the command response.
3. Internal channel path types such as internal-queued-direct-communications (IQD) channels and internal-coupling-peer (ICP) channels are not associated with physical channel paths (PCHIDs).
4. Internal-queued-direct-communications (IQD) channel paths and OSA\_Direct\_Express (OSD) channel paths are the only channel path types that can have network identifiers associated with them.
5. A VIRTUAL version of this command, QUERY VIRTUAL CHPID, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see “QUERY VIRTUAL CHPID” on page 1275.

### Examples

To display all the devices on path 1A, enter:

```
query chpid 1a
```

A typical response is:

```
PATH 1A ONLINE TO DEVICES 0190, 0191, 0193
PATH 1A OFFLINE TO DEVICES 0192
```

To display the channel path type of path 1A, enter:

```
query chpid 1a type
```

A response might be:

```
Path 1A is an ESCON unswitched channel
```

To display the physical channel identifier associated with path 1A, enter:

```
query chpid 1a pchid
```

A response might be:

```
Path 1A is associated with physical channel 061A
```

To display all the devices on path 1A in subchannel set 1, enter:

```
query chpid 1a ss 1
```

A typical response is:

```
Path 1A ONLINE TO DEVICES 10190, 10191, 10193
PATH 1A OFFLINE TO DEVICES 10192
```

### Responses

Response for Path Status:

```
PATH yy {ONLINE } TO DEVICES xxxx(1)
... xxxx(n)
```

```
PATH yy {OFFLINE} TO DEVICES zzzz(1)
... zzzz(n)
```

indicates that path yy to device xxxx(1) through device xxxx(n) is logically available (online), and path yy to device zzzz(1) through device zzzz(n) is logically unavailable (offline).



**yy** is the logical channel path.

**xxxx**

**zzzz**

are the real device numbers.

Response for Path Status when SS operand is specified:

```
PATH yy {ONLINE} TO DEVICES xxxxx(1)
... xxxxx(n)

PATH yy {OFFLINE} TO DEVICES zzzzz(1)
... zzzzz(n)
```

indicates that path yy to device xxxxx(1) through device xxxxx(n) is logically available (online), and path yy to device zzzzz(1) through device zzzzz(n) is logically unavailable (offline).

**yy** is the logical channel path.

**xxxx**

**zzzz**

are the real device numbers.

Response for No Devices on Path:

```
NO DEVICES ON PATH yy
```

indicates there are no devices on the specified channel path.

**yy** is the logical channel path.

Response for Nonexistent CHPID:

```
INVALID CHPID yy
```

indicates that a nonexistent channel path ID was specified.

**yy** is the nonexistent logical channel path.

Response for Quiesced Path:

```
PATH yy QUIESCED TO DEVICE(S) xxxx(1)... xxxx(n)
```

indicates the path has been quiesced (locked offline) to the specified devices by an IBM Service Representative. Once service on this channel path is completed, a corresponding resume request issued by the Service Representative (SR) will cause this channel path to be varied online to the device or devices, if they were online before being quiesced. If for some reason the resume request did not vary this channel path back online, the FORCE option on the VARY PATH command can be used to override the quiesced path and vary the channel path online to the device or devices. The FORCE option should only be used when the SR indicates that service is complete and a resume was issued but did not vary the channel path online.

**yy** is the logical channel path.

**xxxx**

are the real device numbers.

Responses for Optional NETID Operand:

```
PATH NETIDS
yy 0000000000000000 0000000000000000 0000000000000000 0000000000000000
-OR
Path yy is a type of channel path that does not support network identifiers.
```

-OR  
Network identifiers could not be obtained.

**yy**  
is the logical channel path.

**nnnnnnnnnnnnnnnnnn**

is a network identifier. The order of the network identifiers corresponds to the order of the ports associated with the path. Dashes indicate no network identifier has been assigned to the port.

Responses for Optional PCHID Operand:

Path yy is associated with physical channel xxxx  
Path yy is not associated with a physical channel

**yy**  
is the logical channel path.

**xxxx**  
is the physical channel identifier value found.

Responses for Optional TYPE Operand:

Path yy is a parallel block-multiplexer channel  
Path yy is a parallel byte-multiplexer channel  
Path yy is an ESCON unswitched channel  
Path yy is an ESCON switched channel  
Path yy is an ESCON channel  
(switched or unswitched has not been determined)  
Path yy is an ESCON channel to a parallel block-multiplexer converter  
Path yy is a native-interface channel  
Path yy is an ESCON CTC unswitched channel  
Path yy is an ESCON CTC switched channel  
Path yy is an ESCON CTC channel  
(switched or unswitched has not been determined)  
Path yy is a coupling-facility-sender (CFS) channel  
Path yy is a coupling-facility-receiver (CFR) channel  
Path yy is an ESCON channel to a parallel byte-multiplexer converter  
Path yy is an Open Systems Adapter (OSA) channel  
Path yy is an Open Systems Adapter-Express (OSE) channel  
Path yy is an Open Systems Adapter Direct Express (OSD) channel  
Path yy is an Open Systems Adapter 3270-Console (OSC) channel  
Path yy is an Open Systems Adapter NCP (OSN) channel  
Path yy is an internal-system-device (ISD) channel  
Path yy is a cluster-bus-sender (CBS) channel  
Path yy is a cluster-bus-receiver (CBR) channel  
Path yy is an internal-coupling-sender (ICS) channel  
Path yy is an internal-coupling-receiver (ICR) channel  
Path yy is a FICON unswitched channel  
Path yy is a FICON switched channel  
Path yy is a FICON channel to an ESCON Director FICON Bridge card  
Path yy is a FICON channel  
(switched or unswitched has not been determined)  
Path yy is a High Performance FICON unswitched channel  
Path yy is a High Performance FICON switched channel  
Path yy is a High Performance FICON channel  
Path yy is a direct-system-device (DSD) channel  
Path yy is an emulated-I/O (EIO) channel  
Path yy is a cluster-bus-peer (CBP) channel  
Path yy is a coupling-facility-peer (CFP) channel  
Path yy is an internal-coupling-peer (ICP) channel  
Path yy is an internal-queued-direct-communications (IQDC) channel  
Path yy is an open-fibre-channel-protocol (FCP) channel  
Path yy is a coupling over InfiniBand (CIB) channel  
Path yy is an Open Systems Adapter Direct Express IEDN (OSX) channel  
Path yy is an Open Systems Adapter Direct Express INMN (OSM) channel  
Path yy is a coupling over PCIe (CS5) channel  
Path yy is a coupling over RoCE (CL5) channel  
Path yy is an unknown or unrecognized channel type - xx

**yy**  
is the logical channel path.

**xx**  
is the CHPID descriptor value found.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP846E Invalid channel path identifier.
- HCP1006E Command processing cannot complete because the system does not support the logical-channel-subsystem facility.
- HCP1006E Command processing cannot complete because the system does not support the Utility-String facility.

# QUERY CHPIDS



## Authorization

Privilege Class: B, E

## Purpose

Use QUERY CHPIDS to display all 256 of the machine's channel paths and their physical status.

## Usage Notes

1. The QUERY CHPIDS command shows all 256 channel paths on your system and their physical status. For information about logical channel paths, see the QUERY CHPID command (see [“QUERY CHPID” on page 791](#)). For information about logical and physical channel paths associated with a specific device, see the QUERY PATHS command (see [“QUERY PATHS” on page 1064](#)).
2. If a channel path is in single channel service (SCS) mode, it will appear as "not configured" in the display.
3. To change the status of the channel paths in your I/O configuration, use one of the following commands:
  - DEFINE CHPID or PATH (see [“DEFINE CHPID / PATH” on page 132](#))
  - DELETE CHPID or PATH (see [“DELETE CHPID / PATH” on page 259](#))
  - MODIFY CU or CNTLUNIT (see [“MODIFY CU / CNTLUNIT” on page 688](#))
  - VARY CHPID (see [“VARY CHPID” on page 2046](#))

## Responses

Response 1:

To display the status of your machine's channel paths, enter the following:

```

query chpids
  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0x  .  .  .  .  +  .  +  +  +  .  .  .  .  +  +  +
1x  .  +  .  .  +  +  .  .  +  +  +  +  .  +  .  -
2x  .  .  .  .  .  .  .  .  .  .  .  .  .  +  +  +
3x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
4x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
5x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
6x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
7x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
8x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
9x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Ax  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Bx  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Cx  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Dx  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Ex  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Fx  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

+ Available
- Offline
. Not configured
Ready;
    
```

This response shows that there are 18 physical channel paths that can be used (available), 1 channel path that is not available (offline), and 237 channel paths that are not defined (not configured).

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6813E Query CHPIDs command failed with return code *nnnn*.
- HCP6814E Command not valid on this processor.

## QUERY CHPIDV

---

► Query — CHPIDV ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY CHPIDV to display the virtualization directory setting for a user.

### Responses

Response for QUERY CHPIDV:

```
CHPID Virtualization is off
```

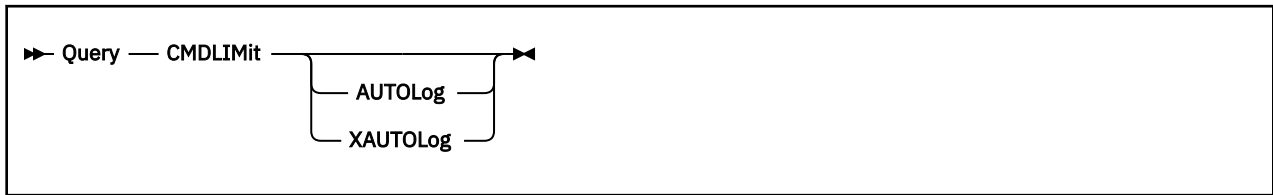
or

```
One path CHPID Virtualization is on
```

### Messages

- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*

## QUERY CMDLIMIT



### Authorization

Privilege Class: A, B

### Purpose

Use QUERY CMDLIMIT to display the number of incorrect passwords the host system accepts.

**Note:** If you exceed this limit, AUTOLOG and XAUTOLOG are disabled for the remainder of your session.

### Operands

#### AUTOLog

#### XAUTOLog

requests a display showing the system limit value for the AUTOLOG and XAUTOLOG commands. This parameter is optional.

### Usage Notes

1. The SET CMDLIMIT command changes the system limit value for the number of incorrect passwords for the AUTOLOG and XAUTOLOG commands.
2. For security reasons, CP limits the number of incorrect passwords allowed in your logon session. Every incorrect password entered for an AUTOLOG or XAUTOLOG command is counted. Once you have exceeded the limit, you must log off and log on again to enter more AUTOLOG or XAUTOLOG commands.
3. AUTOLOG is an accepted synonym for XAUTOLOG. The minimum abbreviation is AUTOL.

### Responses

Response 1:

```
XAUTOLOG/AUTOLOG incorrect password limit is nnn
```

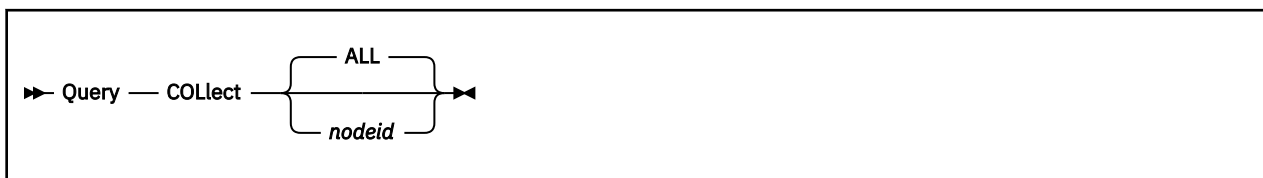
#### **nnn**

indicates the number of incorrect passwords that can be entered before the AUTOLOG and XAUTOLOG commands are disabled for the rest of your logon session.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

## QUERY COLLECT



### Authorization

Privilege Class: G

### Purpose

Use the QUERY COLLECT command to display information about nodes and routes within the ISFC collection.

### Operands

#### ALL

Displays information about the local node and about the immediate neighbors.

#### *nodeid*

Displays the link information for the specified immediate neighbor.

### Responses

Response 1:

```
Node nodeid is local.
```

#### *nodeid*

Specifies the node ID of the local VM node.

Response 2:

```
Node nodeid via link device rdev.
```

#### *nodeid*

Specifies the node ID of the partner node.

#### *rdev*

Specifies the address of the real link device that connects the local node with the partner node.

Response 3:

```
No nodes exist in collection.
```

Response 4:

Up to 16 *rdevs* may be shown for a single node.

```
Node nodeid via link devices rdev1 rdev2 rdev3 rdev4.
```

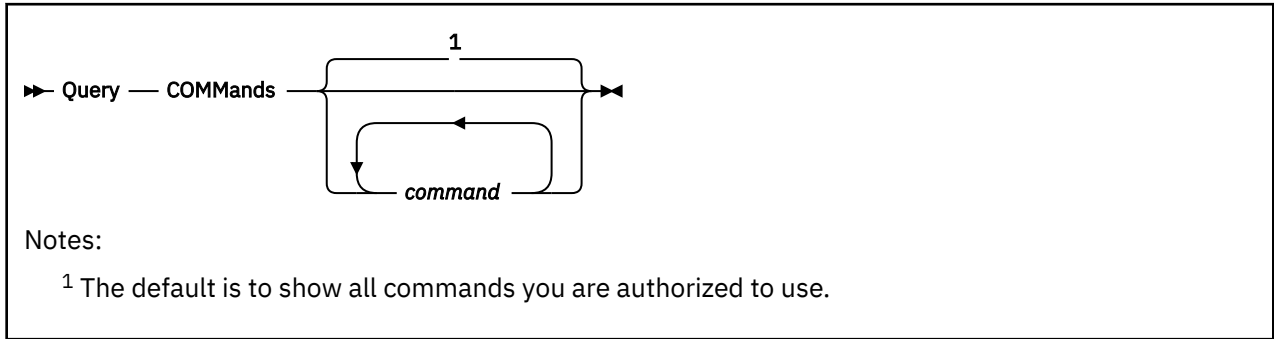
### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*



- HCP2726E Node *nodeid* is not directly connected.

## QUERY COMMANDS



### Authorization

Privilege Class: Any

### Purpose

Use QUERY COMMANDS to display the commands and diagnose codes you are authorized to use.

### Operands

***command***

the names of commands to be checked for authorization.

### Usage Notes

1. The same function is available through the COMMANDS command.
2. Some commands have more than one version, but the user may not be authorized to use all versions of the command. If the system rejects a command, it is probably because the user is not authorized to enter the command with the operand or operands entered.
3. To see the command versions you are authorized to use, enter QUERY COMMANDS *cmd*, where *cmd* is the name of the command or commands that interests you. You are shown the default IBM privilege classes of the command that you are authorized to enter.

You may specify any number of commands. If a command name is entered twice, it appears twice in the response. No duplicate-entry checking is done.

4. Diagnose codes cannot be queried separately from commands.
5. IBMCLASS=ANY will output if:
  - the command is allowed before logon and not disabled, or
  - the user privilege class is ANY, meaning the command version is not associated with any particular privilege class and the command is enabled.

### Responses

Response 1:

```
cmd1    cmd2    cmd3    cmd4    cmd5    cmd6
cmd7    ...
:
diag1   diag2   diag3   diag4   diag5   diag6
diag7   ...
```

is the response when the command QUERY COMMANDS is entered alone; all authorized commands and diagnose codes are displayed.

**cmd1,2...**

the names of authorized commands.

**diag1,2...**

the names of authorized diagnose codes.

Response 2:

```
AUTHORIZED {QUERY|SET} OPERANDS:
operand1  IBMCLASS={c1,c2,c3}
           {ANY}
           {NONE}
operand3  IBMCLASS={c1,c2,c3}
           {ANY}
           {NONE}
:
operand2  IBMCLASS={c1,c2,c3}
           {ANY}
           {NONE}
operand4  IBMCLASS={c1,c2,c3}
           {ANY}
           {NONE}
```

is the response when the command COMMANDS is entered with the QUERY or SET operand; all authorized operands of the QUERY or SET command are displayed.

**operand1,2 ...**

the names of authorized QUERY or SET operands.

**c1,c2, ...**

the IBM-defined privilege classes of the operands that you are authorized to enter.

Response 3:

```
cmd1      IBMCLASS=c1,c2,c3
cmd2      IBMCLASS=ANY
cmd3      IBMCLASS=NONE
```

is the response you receive when you specifically enter commands to check authorization for them. The response includes one line of output for each command listed.

**cmdx**

is the name of the command you entered on the command line.

**IBMCLASS=c1,c2,c3**

the IBM-defined privilege classes of the command specified; the privilege class displayed reflects the authorization of the user making the inquiry.

**IBMCLASS=ANY**

indicates that the command is a class ANY command, and all users may enter the command.

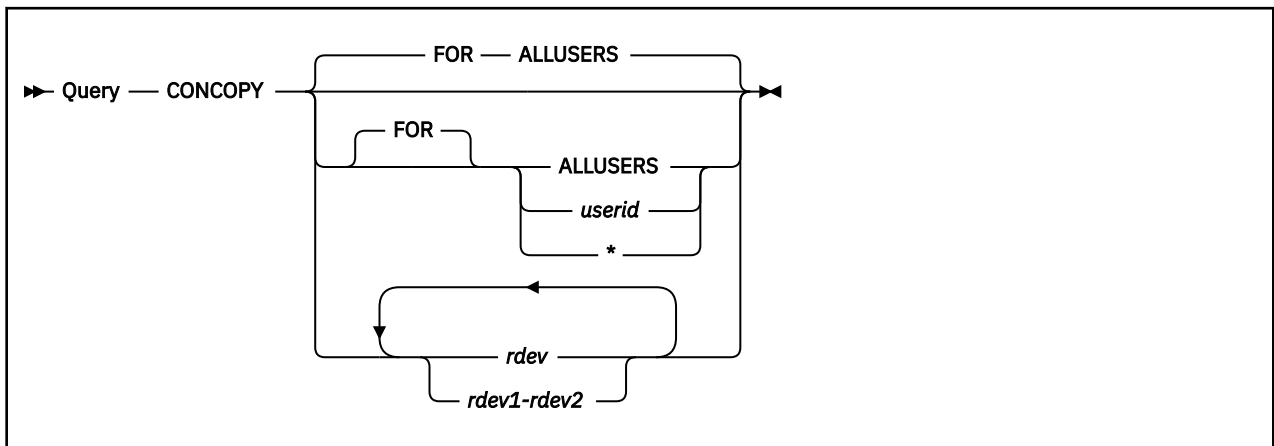
**IBMCLASS=NONE**

indicates that you are not authorized to enter any version of the command.

**Messages**

- HCP003E Invalid option - {option|command contains extra option(s) starting with option}

## QUERY CONCOPY



### Authorization

Privilege Class: B

### Purpose

Use the QUERY CONCOPY command to display Concurrent Copy (CONCOPY) session information.

### Operands

#### ALLUSERS

indicates that the query operation applies to all the virtual machines. This is the default.

#### userid

is the virtual machine for which CONCOPY session information is to be displayed.

\*

indicates that the query operation applies to the virtual machine issuing the command.

#### rdev

#### rdev1-rdev2

identifies the real device number of one of the subsystem devices. You can specify a single device number, a list of device numbers, or a range of device numbers.

### Examples

Example 1:

To find out what users on your system have active CONCOPY sessions, enter:

```
query concopy
```

You receive Response 1 for each user who has an active CONCOPY session. If your system has three such users (ADAMS, BAKER, and DUNCAN), the display might look like this:

```
CONCOPY session(s) for userid ADAMS are on
device(s): 301 321
CONCOPY session(s) for userid BAKER are on
device(s): 333 433 533
CONCOPY session(s) for userid DUNCAN are on
device(s): 321 322
```

Example 2:

To get the CONCOPY status of all devices on your system in the device number range from 320 to 325, enter:

```
query concopy 320-325
```

For each device in the specified range, you receive a CONCOPY status response (Response 2 or Response 5) if the device exists, or an error message if the device does not exist. If your system has three devices in that range (321, 322, and 323), and two (321 and 322) are included in active CONCOPY sessions, the display might look like this:

```
HCPnnn040E Device 320 does not exist
CONCOPY session(s) on device 321 for userid(s):
  ADAMS DUNCAN
CONCOPY session(s) on device 322 for userid(s):
  DUNCAN
No CONCOPY session exists on device 323
HCPnnn040E Device 324 does not exist
HCPnnn040E Device 325 does not exist
```

## Responses

Response 1:

```
CONCOPY session(s) for userid userid are on
device(s): rdev1...
```

indicates that virtual machine *userid* has active CONCOPY sessions and lists the real device numbers of the devices that are part of the CONCOPY sessions.

Response 2:

```
CONCOPY session(s) on device rdev for userid(s):
userid...
```

indicates that real device *rdev* is part of at least one CONCOPY session for the specified user ID or list of user IDs.

Response 3:

```
No CONCOPY session exists for userid userid.
```

indicates that virtual machine *userid* does not have any CONCOPY sessions.

Response 4:

```
No CONCOPY session exists for any user.
```

indicates that there are no CONCOPY sessions for any user on any storage control.

Response 5:

```
No CONCOPY session exists on device rdev.
```

indicates that device *rdev* is not part of any CONCOPY session.

## Messages

- HCP006E Invalid device type - *rdev*
- HCP009E Invalid range - *range*
- HCP026E Operand missing or invalid
- HCP040E Device *rdev* does not exist
- HCP046E *type rdev* offline
- HCP332E Invalid control unit type - *rdev*

## QUERY CONFIGMODE

► Query — CONFIGmode ◄

### Authorization

Privilege Class: B, E

### Purpose

Use QUERY CONFIGMODE to display whether the processor is currently executing commands to dynamically change its I/O configuration.

**Note:** QUERY CONFIGMODE is a dynamic I/O command. Dynamic I/O configuration is only supported on IBM Z servers. If your system is not running on one of these servers, your command will not complete successfully.

### Usage Notes

1. To turn configuration mode on or off, use the SET CONFIGMODE command (see [“SET CONFIGMODE” on page 1436](#)).
2. Whenever you issue one of the following dynamic I/O commands:

DEFINE CHPID	DEFINE PATH
DELETE CHPID	DELETE PATH
MODIFY CHPID	MODIFY PATH
DEFINE CU	DEFINE CNTLUNIT
DELETE CU	DELETE CNTLUNIT
MODIFY CU	MODIFY CNTLUNIT
DEFINE DEVICE	DEFINE IODEVICE
DELETE DEVICE	DELETE IODEVICE
MODIFY DEVICE	MODIFY IODEVICE
DEFINE LPAR	
DELETE LPAR	
SET LSYSTEM	

CP puts the machine in configuration mode, processes the dynamic I/O command, and then takes you out of configuration mode.

3. Whenever you dynamically change your I/O configuration, CP must perform a certain amount of overhead. If you want to issue several dynamic I/O commands in succession and you are concerned about performance, issue the SET CONFIGMODE ON command, issue your dynamic I/O commands, and then issue the SET CONFIGMODE OFF command. This method causes CP to perform the overhead only while processing the SET CONFIGMODE commands, rather than before and after each individual command.

Note that the SET LSYSTEM command is an exception to the above. CP will not process this command while the machine is in configuration mode. If you want to issue the SET LSYSTEM command to set or change the current local system name, you must exit configuration mode and enter the SET LSYSTEM command as an individual command. For more information about the SET LSYSTEM command, see [“SET LSYSTEM” on page 1553](#).

4. While you are in configuration mode, CP will not process any SET DYNAMIC\_I/O OFF commands. If you want to prevent people from making dynamic I/O changes, you must exit configuration mode and enter the SET DYNAMIC\_I/O OFF command as an individual command. For more information about the SET DYNAMIC\_I/O command, see [“SET DYNAMIC\\_I/O” on page 1502](#).

## Responses

Response when Configuration Mode is Off:

To display whether your processor is currently in configuration mode, enter the following:

```
query configmode
System is not in configuration mode.
Ready;
```

This response tells you that your processor is not currently executing commands to dynamically change its I/O configuration.

Response when Configuration Mode is Turned On:

If someone on your system had turned configuration mode on using the SET CONFIGMODE command and was in the process of issuing one or more dynamic I/O commands, you would see the response below. You would also see this response if someone on your system had issued a dynamic I/O command that caused the system to temporarily turn on configuration mode:

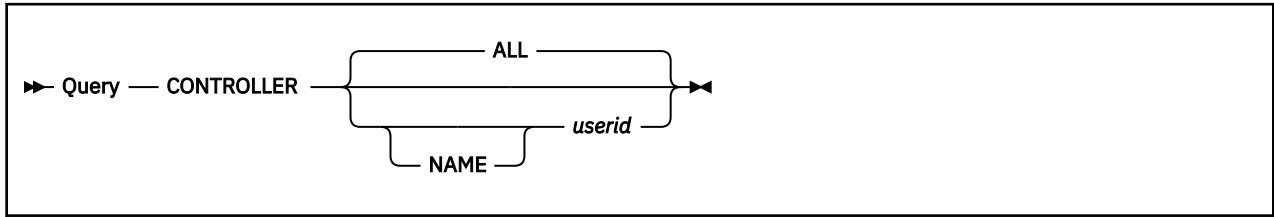
```
query configmode
System is in configuration mode for user userid
Ready;
```

Where *userid* is the name of the person who issued the SET CONFIGMODE command.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6808E Cannot find the configuration token on the system
- HCP6808E Cannot store LPAR data
- HCP6808E System disabled for dynamic I/O changes
- HCP6808E The configuration mode setting of the hardware and software do not match
- HCP6808E The dynamic I/O support was disabled at the service console
- HCP6808E The VM configuration token does not match the hardware configuration token
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed on this system

## QUERY CONTROLLER



### Authorization

Privilege Class: B

### Purpose

Use QUERY CONTROLLER to display information about the z/VM TCP/IP virtual machines used to manage OSA-Express or HiperSockets Bridge Port devices associated with virtual switches.

### Operands

#### ALL

displays information for all possible virtual switch controllers. This is the default value when an explicit *NAME userid* is omitted.

#### NAME *userid*

restricts the response to the virtual switch controller specified.

### Examples

This example displays all the available detail for CONTROLLER.

```
query controller
Controller DTCVSW1 Available: YES VDEV Range: 8800-88FF Level 630
Capability: IP ETHERNET VLAN_ARP GVRP LINKAGG ISOLATION
BRIDGE_CAPABLE VEPA SHARED_LINKAGG PRIQUEUING
SYSTEM EDUC1 Active Controller: DTCVSW1 RDEV: 8803
SYSTEM TECHNET1 Backup Controller: * RDEV: 8800

Controller DTCVSW2 Available: YES VDEV Range: * Level 630
Capability: IP ETHERNET VLAN_ARP GVRP LINKAGG ISOLATION
BRIDGE_CAPABLE VEPA SHARED_LINKAGG PRIQUEUING

SYSTEM TECHNET1 Active Controller: * RDEV: 1B00

Controller TCPIP1 Available: YES VDEV Range: * Level 630 Failover disabled
Capability: IP ETHERNET VLAN_ARP GVRP LINKAGG ISOLATION
BRIDGE_CAPABLE VEPA SHARED_LINKAGG PRIQUEUING
SYSTEM PAYROLL Active Controller: * VDEV: 2B03
SYSTEM TECHNET1 Backup Controller: * VDEV: 3B00

Controller TCPIP3 Available: YES VDEV Range: * Level 630 Stalled
Capability: IP ETHERNET VLAN_ARP GVRP LINKAGG ISOLATION
BRIDGE_CAPABLE VEPA SHARED_LINKAGG PRIQUEUING
```

### Responses

Each virtual switch that is currently managed is listed, along with the current setting for the Controller option.

#### Available

indicates if the virtual switch controller is available to control an additional set of OSA-Express or HiperSockets Bridge Port devices associated with the virtual switch.



**VDEV Range**

identifies the device range where the OSA-Express or HiperSockets Bridge Port devices associated with a virtual switch can be attached. **VDEV Range: \*** indicates that the virtual device address used to attach the devices is the same as the real device address identified by the virtual switch's RDEV option.

**LEVEL**

indicates the level of the TCP/IP stack running in the controller virtual machine. "Initializing" is displayed if the level is not yet known.

**Failover disabled**

indicates that the TCP/IP stack machine has been configured so that CP does not perform timestamp checking. Timestamp checking is used to confirm that the TCP/IP stack is responding, and it has been disabled in this instance.

**Stalled**

indicates the TCP/IP stack machine is not responding in a timely manner. Use the `FAILOVER_DISABLED` option on the TCP/IP VSWITCH CONTROLLER configuration statement if you need to do tracing or debug for the TCP/IP stack machine that might result in slow response time. See *z/VM: TCP/IP Planning and Customization* for additional information.

**Capability**

indicates what types of virtual switches the controller can manage.

**IP and ETHERNET**

The virtual switch controller can initialize an OSA-Express device in IP or ETHERNET modes.

**VLAN\_ARP**

It can register IP addresses on the OSA-Express with the proper VLAN groups.

**GVRP**

It can register VLAN IDs in use on a virtual switch with GVRP-aware switches (GVRP).

**LINKAGG**

Virtual switches that use link aggregation can be managed.

**ISOLATION**

Virtual switches that use the isolation setting can be managed.

**BRIDGE\_CAPABLE**

The controller can manage a virtual switch configured with a HiperSocket bridge port.

**VEPA**

The controller can manage virtual switches that use the VEPA option on SET VSWITCH.

**SHARED\_LINKAGG**

Virtual switches configured with shared port groups can be managed.

**PRIQUEUING**

Virtual switches configured with priority queuing can be managed.

**Active**

indicates that the device is the current active connection for the virtual switch.

**Backup**

indicates that some initialization has been done so that error recovery can happen more quickly if the active device has a problem.

**[Active | Standby | Undefined | Inactive] Bridge**

indicates that the device is operating as a virtual switch Bridge Port connection.

Typically, the bridge status will be Active or Standby. Undefined and Inactive are expected to be temporary while the port is in transition.

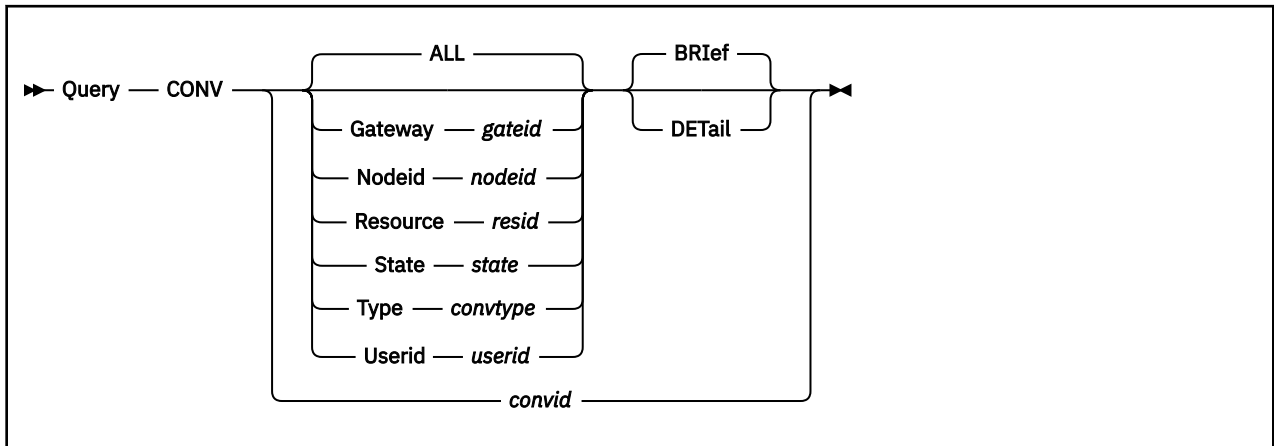
**Controller: \***

indicates that a different z/VM TCP/IP stack takes over in case there is a problem with the current virtual switch controller.

**RDEV: devno**

shows the base real device number attached to the controller for the virtual switch.

## QUERY CONV



### Authorization

Privilege Class: B

### Purpose

Use the QUERY CONV command to display information about the conversations that are managed by the local VM node.

### Operands

#### ALL

displays information for all active conversations that are in the domain of the local VM node. ALL is the default.

#### Gateway *gateid*

displays information for the active conversations that are established through gateways such as, APPC/VM VTAM Support (AVS) or system gateways defined on VM systems in the Communication Services (CS) collection.

#### Nodeid *nodeid*

displays information for the active conversations that are established between the specified remote node and the local VM node.

#### Resource *resid*

displays information for the active conversations that are established for the application *resid*. This application is a global resource.

#### State *state*

displays information for the active conversations that are in the specified state. Valid states are CONFIRM, CONNECT, PENDING, RECEIVE, RESET, SEND, and SEVER.

#### Type *convtype*

displays information about active conversations of the specified type; valid types are APPC, IUCV, CMP, or CPAP.

#### Userid *userid*

displays information for the active conversations established for the user *userid*. This user ID is either:

- The access security user ID specified in the connection request
- The user ID of the owner of a private resource. This private resource is located in a virtual machine on the local z/VM system or on a workstation in the domain of the local VM node.

**BRIef**

displays the number of active conversations. BRIef is the default.

**DETail**

displays detailed information about each conversation.

**convid**

is the hexadecimal conversation identifier of a specific conversation managed by the local VM node.

**Responses**

Response 1:

```
Conversation: convid   Instance: x       Type:           APPC
Node:    nodeid       Bytes Sent:    nnn
State:   state        Bytes Received: nnn
Access Userid: userid Gateway Name:  gateid
```

**convid**

specifies the conversation ID assigned by the local VM node.

**x**

specifies the conversation instance number.

**APPC**

specifies that the conversation is APPC.

**nodeid**

specifies the node ID of the remote node.

**nnn**

specifies the number of bytes sent from the local node or the number of bytes received by the local node.

**state**

specifies if the conversation is in CONFIRM, CONNECT, PENDING, RECEIVE, RESET, SEND, or SEVER state.

**userid**

specifies the access user ID that establishes the conversation.

**gateid**

specifies the system or AVS gateway through which this conversation is allocated.

Response 2:

```
Conversation: convid   Instance: x       Type:           APPC
Node:    nodeid       Bytes Sent:    nnn
State:   state        Bytes Received: nnn
Access Userid: userid Resource Name:  resid
```

**convid**

specifies the conversation ID assigned by the local VM node.

**x**

specifies the conversation instance number.

**APPC**

specifies that the conversation is APPC.

**nodeid**

specifies the node ID of the remote node.

**nnn**

specifies the number of bytes sent from the local node or the number of bytes received by the local node.

**state**

specifies if the conversation is in CONFIRM, CONNECT, PENDING, RECEIVE, RESET, SEND, or SEVER state.

## QUERY CONV

### **userid**

specifies the access user ID that establishes the conversation.

### **resid**

specifies the resource this conversation is connected to.

Response 3:

```
Conversation: convid      Instance: x      Type:          APPC
Node:      nodeid
State     state          Bytes Sent:    nnn
Access Userid: userid    Bytes Received: nnn
Target Userid: targetid
```

### **convid**

specifies the conversation ID assigned by the local VM node.

### **x**

specifies the conversation instance number.

### **APPC**

specifies that the conversation is APPC.

### **nodeid**

specifies the node ID of the remote node.

### **nnn**

specifies the number of bytes sent from the local node or the number of bytes received by the local node.

### **state**

specifies if the conversation is in CONFIRM, CONNECT, PENDING, RECEIVE, RESET, SEND, or SEVER state.

### **userid**

specifies the access user ID that establishes the conversation.

### **targetid**

specifies the target user ID this conversation is connected to.

Response 4:

```
Conversation: convid      Instance: x      Type:          convtype
Node:      nodeid
Bytes Sent:    nnn
Bytes Received: nnn
```

### **convid**

specifies the conversation ID assigned by the local VM node.

### **x**

specifies the conversation instance number.

### **convtype**

specifies if the conversation is APPC, IUCV, CMP, or CPAP.

### **nodeid**

specifies the node ID of the remote node.

### **nnn**

specifies the number of bytes sent from the local node or the number of bytes received by the local node.

Response 5:

```
nnn conversations active.
```

### **nnn**

specifies the number of conversation(s) active.

Response 6:

```
nnn conversation(s) active with user userid
```

**nnn**

specifies the number of conversations active.

**userid**

specifies the access user ID that establishes the conversation.

Response 7:

```
nnn conversation(s) active with node nodeid
```

**nnn**

specifies the number of conversations active.

**nodeid**

specifies the node ID of the remote node.

Response 8:

```
nnn conversation(s) active with resource resid
```

**nnn**

specifies the number of conversations active.

**resid**

specifies the resource to which this conversation is connected.

Response 9:

```
nnn conversation(s) active with gateway gateid
```

**nnn**

specifies the number of conversations active.

**gateid**

specifies the system or AVS gateway through which this conversation is allocated.

Response 10:

```
nnn conversation(s) active of type convtype
```

**nnn**

specifies the number of conversations active.

**convtype**

specifies the type of conversation.

Response 11:

```
nnn conversation(s) currently in state state
```

**nnn**

specifies the number of conversations active.

**state**

specifies the state of the conversation(s).

Response 12:

```
Conversation: convid Instance: x Type: APPC
Node: nodeid Bytes Sent: nnn
PASSTHRU Bytes Received: nnn
Access Userid: userid Gateway Name: gateid
```

**convid**

specifies the conversation ID assigned by the local VM node.

## QUERY CONV

**x**

specifies the conversation instance number.

**APPC**

specifies that the conversation is APPC.

**nodeid**

specifies the node ID of the remote node.

**PASSTHRU**

indicates that this is an intermediate node for the conversation.

**nnn**

Bytes sent specifies the number of bytes transmitted from the target node to the source node by the local node.

Bytes received specifies the number of bytes transmitted from the source node to the target node by the local node.

**userid**

specifies the access user ID that establishes the conversation.

**gateid**

specifies the system or AVS gateway through which this conversation is allocated.

Response 13:

```
Conversation: convid      Instance: x      Type:          APPC
Node:      nodeid      Bytes Sent:    nnn
PASSTHRU   Bytes Received: nnn
Access Userid: userid  Resource Name: resid
```

**convid**

specifies the conversation ID assigned by the local VM node.

**x**

specifies the conversation instance number.

**APPC**

specifies that the conversation is APPC.

**nodeid**

specifies the node ID of the remote node.

**PASSTHRU**

indicates that this is an intermediate node for the conversation.

**nnn**

Bytes sent specifies the number of bytes transmitted from the target node to the source node by the local node.

Bytes received specifies the number of bytes transmitted from the source node to the target node by the local node.

**userid**

specifies the access user ID that establishes the conversation.

**resid**

specifies the resource this conversation is connected to.

Response 14:

```
Conversation: convid      Instance: x      Type:          APPC
Node:      nodeid      Bytes Sent:    nnn
PASSTHRU   Bytes Received: nnn
Access Userid: userid  Target Userid: targetid
```

**convid**

specifies the conversation ID assigned by the local VM node.

**x**

specifies the conversation instance number.

**APPC**

specifies that the conversation is APPC.

**nodeid**

specifies the node ID of the remote node.

**PASSTHRU**

indicates that this is an intermediate node for the conversation.

**nnn**

Bytes sent specifies the number of bytes transmitted from the target node to the source node by the local node.

Bytes received specifies the number of bytes transmitted from the source node to the target node by the local node.

**userid**

specifies the access user ID that establishes the conversation.

**targetid**

specifies the target user ID this conversation is connected to.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP2728E Conversation *convid* does not exist.

## QUERY CPCHECKING

---

► Query — CPCHECKing ◄

### Authorization

Privilege Class: A, C, E

### Purpose

Use QUERY CPCHECKING to display the status of internal CP checking.

### Responses

```
CPCHECKING=ON {ABEND|VMSTOP }  
OFF
```

#### ABEND

indicates that CP internal checking is activated and an abend should occur if CP internal checking finds an error. When it finds an error it produces an ASR001 abend.

#### VMSTOP

indicates that CP internal checking is activated and if any of the conditions specified by CP checking are untrue, CP issues a Diagnose X'08' instruction with a command string length of zero, causing the virtual machine to stop and CP to post a read to the console.

#### OFF

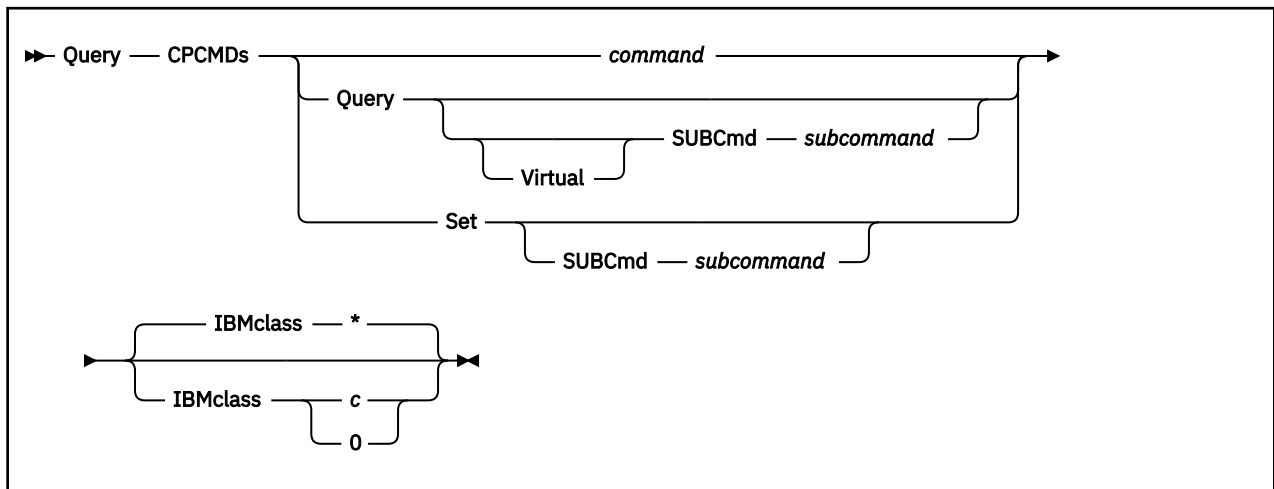
indicates that CP internal checking is deactivated.

### Messages

- HCP1007E The VMSTOP option is only permitted when CP itself is running in a virtual machine.
- HCP6010I The CPCHECKING setting has been changed to OFF to avoid a possible abend loop. It may be set ON again if desired.



## QUERY CPCMDS



### Authorization

Privilege Class: A, C, E

### Purpose

Use QUERY CPCMDS to query the status and display information about a specified CP command. This information includes:

- Enabled/disabled status
- Name of the entry point to handle the command
- IBM Class
- Current User Privilege class of the command
- Address of the CP command table entry block

### Operands

#### **command**

tells CP to query the specified CP command. The variable *command* is any valid CP command, any valid abbreviation of that command, or any character string (including the pattern matching characters, \* and %) representing that command. For more information about minimum abbreviations, see Usage Note [“2”](#) on page 818.

#### **Query**

tells CP to display information about the root CP QUERY command. To query a specific QUERY subcommand, specify the SUBCMD operand.

#### **Virtual**

tells CP that you are querying a QUERY VIRTUAL subcommand.

#### **SUBCmd subcommand**

specifies the name of the QUERY or SET subcommand whose information you want to display. The variable *subcommand* is any valid QUERY or SET subcommand, any valid abbreviation of that subcommand, or any character string (including the pattern matching characters, \* and %) representing that subcommand. For more information about minimum abbreviations, see Usage Note [“2”](#) on page 818.

**Set**

tells CP to display information about the root CP SET command. To query a specific SET subcommand, specify the SUBCMD operand.

**IBMclass \***

specifies that you want to display information for all the IBM classes associated with the specified CP command. If omitted, IBMCLASS \* is the default.

**IBMclass c**

specifies that you only want to display information for the specific IBM class associated with the specified CP command. The variable c is the IBM-defined privilege class A, B, C, D, E, F, or G.

**IBMclass 0**

specifies that you only want to display information for CP commands with no assigned IBM class. Only commands originally defined with privilege class "ANY" have no assigned IBM class.

**Usage Notes**

1. For each command, CP has at least one command table entry block. If the command has more than one IBM class, CP has one command table entry block for each version of the command. For example, the DEFINE command has four IBM classes: A, B, E, and G. To display information for the DEFINE GRAF command, you want the command table entry block for the class G DEFINE command:

```
query cpcmds define ibmclass g
```

The only exceptions to this are the QUERY and SET commands. CP has at least one command table entry block for each QUERY and SET subcommand. To display information for the QUERY TIMEZONES subcommand, you would issue one of the following commands:

```
query cpcmds query subcmd timezones
query cpcmds query subcmd timezones ibmclass g
```

2. You can use the minimum abbreviation of the command or subcommand name. For example, issuing either of the following commands produces the same results:

```
query cpcmds query subcmd privclass
query cpcmds q subcmd priv
```

3. To define a new CP command or a new version (by IBM class) of an existing CP command, use the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD”](#) on page 143).
4. To define a new alias for an existing CP command, use the DEFINE ALIAS command (see [“DEFINE ALIAS”](#) on page 127).
5. To load the command processing code into the system execution space, use the CPXLOAD command (see [“CPXLOAD”](#) on page 111).
6. To activate a CP command:
  - While defining it, use the ENABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD”](#) on page 143).
  - After defining it, use the ENABLE COMMAND or CMD command (see [“ENABLE COMMAND / CMD”](#) on page 460).
7. To change the definition of an existing CP command, use the MODIFY COMMAND command (see [“MODIFY COMMAND / CMD”](#) on page 683).
8. To deactivate a CP command:
  - While defining it, use the DISABLE operand of the DEFINE COMMAND or CMD command (see [“DEFINE COMMAND / CMD”](#) on page 143).
  - After defining it, use the DISABLE COMMAND or CMD command (see [“DISABLE COMMAND / CMD”](#) on page 313).
9. To remove the command processing code from the system execution space, use the CPXUNLOAD command (see [“CPXUNLOAD”](#) on page 117).

10. For more information about user-defined commands, see the [Introduction](#) in *z/VM: CP Exit Customization*.

## Responses

Response 1:

To query the CP MESSAGE command, enter the following:

```
query cpcmds message
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   A            PrivClasses: A
  CMDBK Address: 001CCF98   Entry Point: HCPXMGMS
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   B            PrivClasses: B
  CMDBK Address: 001CD008   Entry Point: HCPXMGMS
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   0            PrivClasses: <ANY>
  CMDBK Address: 001CD078   Entry Point: HCPXMGMS
Ready;
```

In this example,

### Command

shows the name of the command. When querying a QUERY or SET subcommand, the response will include the full subcommand name. See Response 5 for an example.

### Status

tells you whether the command is enabled or disabled and if the SILENTLY option is in effect.

### IBM Class

shows the IBM class of the command. The possible values are:

- A** this is a system-control command to be used by the primary system operator.
- B** this is a command for operational control of real devices.
- C** this is a command to alter system storage.
- D** this is a command for system-wide control of spool files.
- E** this is a command to examine system storage.
- F** this is a command for service control of real devices.
- G** this is a general-use command used to control the functions of a virtual machine.
- 0** (zero) this command has no specific IBM class assigned.

### PrivClasses

shows the privilege class (or classes) of the command. This usually contains 1 or more privilege classes in the range A to Z and 1 to 6. It may also contain "ANY" which indicates that this command can be issued by users with any privilege class.

### CMDBK Address

shows the host logical storage address of the command table entry block for the command.

### Entry Point

shows the name of the entry point which will handle this command.

Response 2:

The CP MSG command is actually an alias for the CP MESSAGE command. CP displays different information for an alias than it does for a command. For example, to display information for the CP MSG command alias, enter the following:

```
query cpcmds msg
Alias: MSG
  Alias Status: Enabled      Not Silent
  Alias CMDBK Address: 001CD388
  Command: MESSAGE
  IBM Class:      A          PrivClasses: A
  CMDBK Address: 001CCF98    Entry Point: HCPXMGMS
Alias: MSG
  Alias Status: Enabled      Not Silent
  Alias CMDBK Address: 001CD3F8
  Command: MESSAGE
  IBM Class:      B          PrivClasses: B
  CMDBK Address: 001CD008    Entry Point: HCPXMGMS
Alias: MSG
  Alias Status: Enabled      Not Silent
  Alias CMDBK Address: 001CD468
  Command: MESSAGE
  IBM Class:      0          PrivClasses: <ANY>
  CMDBK Address: 001CD078    Entry Point: HCPXMGMS
Ready;
```

In this example,

### **Alias**

shows the name of the command and indicates that command is an alias of another. When querying a QUERY or SET subcommand, the response will include the full subcommand name.

### **Alias Status**

tells you whether the alias is enabled or disabled and if the SILENTLY option is in effect.

### **Alias CMDBK Address**

shows the host logical storage address of the command table entry block for the alias.

### **Command**

shows the name of the command that has been aliased. When querying a QUERY or SET subcommand, the response will include the full subcommand name.

### **IBM Class**

shows the IBM class of the command. The possible values are:

#### **A**

this is a system-control command to be used by the primary system operator.

#### **B**

this is a command for operational control of real devices.

#### **C**

this is a command to alter system storage.

#### **D**

this is a command for system-wide control of spool files.

#### **E**

this is a command to examine system storage.

#### **F**

this is a command for service control of real devices.

#### **G**

this is a general-use command used to control the functions of a virtual machine.

#### **O**

(zero) this command has no specific IBM class assigned.

**PrivClasses**

shows the privilege class (or classes) of the command. This usually contains 1 or more privilege classes in the range A to Z and 1 to 6. It may also contain "ANY" which indicates that this command can be issued by users with any privilege class.

**CMDBK Address**

shows the host logical storage address of the command table entry block for the command that has been aliased.

**Entry Point**

shows the name of the entry point which will handle this command.

Response 3:

The CP MESSAGE and MSG commands also share the same minimum abbreviation, M. This means you can display information about both commands by specifying their common minimum abbreviation:

```

query cpcmds m
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   A            PrivClasses: A
  CMDBK Address: 001CCF98   Entry Point: HCPXMGMS
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   B            PrivClasses: B
  CMDBK Address: 001CD008   Entry Point: HCPXMGMS
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   0            PrivClasses: <ANY>
  CMDBK Address: 001CD078   Entry Point: HCPXMGMS
Alias: MSG
  Alias Status: Enabled     Not Silent
  Alias CMDBK Address: 001CD388
  Command: MESSAGE
  IBM Class:   A            PrivClasses: A
  CMDBK Address: 001CCF98   Entry Point: HCPXMGMS
Alias: MSG
  Alias Status: Enabled     Not Silent
  Alias CMDBK Address: 001CD3F8
  Command: MESSAGE
  IBM Class:   B            PrivClasses: B
  CMDBK Address: 001CD008   Entry Point: HCPXMGMS
Alias: MSG
  Alias Status: Enabled     Not Silent
  Alias CMDBK Address: 001CD468
  Command: MESSAGE
  IBM Class:   0            PrivClasses: <ANY>
  CMDBK Address: 001CD078   Entry Point: HCPXMGMS

```

Response 4:

To display information for all CP commands whose name begins with the letter M, enter the following:

```

query cpcmds m*
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   A            PrivClasses: A
  CMDBK Address: 001CCF98   Entry Point: HCPXMGMS
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   B            PrivClasses: B
  CMDBK Address: 001CD008   Entry Point: HCPXMGMS
Command: MESSAGE
  Status:      Enabled      Not Silent
  IBM Class:   0            PrivClasses: <ANY>
  CMDBK Address: 001CD078   Entry Point: HCPXMGMS
Command: MIGRATE
  Status:      Enabled      Not Silent
  IBM Class:   A            PrivClasses: A
  CMDBK Address: 001CD0E8   Entry Point: HCPNOSCM
Command: MODIFY
  Status:      Enabled      Not Silent
  IBM Class:   A            PrivClasses: A
  CMDBK Address: 001CD158   Entry Point: HCPRSCOV
Command: MODIFY
  Status:      Enabled      Not Silent
  IBM Class:   B            PrivClasses: B

```

## QUERY CPCMD5

```
CMDBK Address: 001CD1C8   Entry Point: HCPRSCOV
Command: MONITOR
Status:      Enabled      Not Silent
IBM Class:   A            PrivClasses: M
CMDBK Address: 001CD238   Entry Point: HCPMNCCM
Command: MONITOR
Status:      Enabled      Not Silent
IBM Class:   E            PrivClasses: S
CMDBK Address: 001CD2A8   Entry Point: HCPMNCCM
Command: MSGNOH
Status:      Enabled      Not Silent
IBM Class:   B            PrivClasses: KRZ
CMDBK Address: 001CD318   Entry Point: HCPXMGNH
Alias: MSG
Alias Status: Enabled      Not Silent
Alias CMDBK Address: 001CD388
Command: MESSAGE
IBM Class:   A            PrivClasses: A
CMDBK Address: 001CCF98   Entry Point: HCPXMGMS
Alias: MSG
Alias Status: Enabled      Not Silent
Alias CMDBK Address: 001CD3F8
Command: MESSAGE
IBM Class:   B            PrivClasses: B
CMDBK Address: 001CD008   Entry Point: HCPXMGMS
Alias: MSG
Alias Status: Enabled      Not Silent
Alias CMDBK Address: 001CD468
Command: MESSAGE
IBM Class:   0            PrivClasses: <ANY>
CMDBK Address: 001CD078   Entry Point: HCPXMGMS
Ready;
```

In this example, note that the classes for the MONITOR and MSGNOH commands were redefined by this installation through user MODIFY commands.

Response 5:

To display information for the SET PRIVCLASS subcommand, enter the following:

```
query cpcmds set subcmd privclass
Command: SET PRIVCLASS
Status:      Enabled      Not Silent
IBM Class:   C            PrivClasses: C
CMDBK Address: 002338A8   Entry Point: HCPPCLSE
Command: SET PRIVCLASS
Status:      Enabled      Not Silent
IBM Class:   0            PrivClasses: <ANY>
CMDBK Address: 00233918   Entry Point: HCPPCLSE
Ready;
```

Response 6:

If someone had used a MODIFY COMMAND command or configuration file statement to override an existing CP DISABLE command, you would see something like this:

```
query cpcmds disable
Command: DISABLE
Status:      Enabled      Not Silent
IBM Class:   A            PrivClasses: A
CMDBK Address: 001CBB08   Entry Point: HCPRSCDI
Command: DISABLE
Status:      Enabled      Not Silent
IBM Class:   B            PrivClasses: Z6
CMDBK Address: 001CBB78   Entry Point: HCPRSCDI
Command: -----
Status:      Enabled      Not Silent
IBM Class:   B            PrivClasses: B
CMDBK Address: 0162F348   Entry Point: HCPRSCDI
Command: DISABLE
Status:      Enabled      Not Silent
IBM Class:   F            PrivClasses: F
CMDBK Address: 001CBBE8   Entry Point: HCPRSCDI
Ready;
```

This response shows you there are command table entry blocks for four DISABLE commands: one for IBM class A, two for IBM class B, and one for IBM class F. The first IBM class B DISABLE command is the one that is active. The second IBM class B DISABLE command was the original IBM class B command.

To return to the original IBM class B DISABLE command, enter the following:

```
modify command disable ibmclass b reset
```

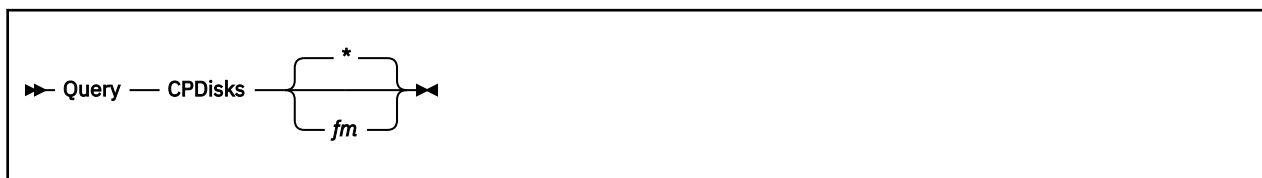
This causes the first IBM class B DISABLE command to disappear from the display and changes the ----- to DISABLE. Thus, the response would look like this:

```
query cpcmds disable
Command: DISABLE
  Status:      Enabled      Not Silent
  IBM Class:   A            PrivClasses: A
  CMDBK Address: 001CBB08   Entry Point: HCPRSCDI
Command: DISABLE
  Status:      Enabled      Not Silent
  IBM Class:   B            PrivClasses: Z6
  CMDBK Address: 001CBB78   Entry Point: HCPRSCDI
Command: DISABLE
  Status:      Enabled      Not Silent
  IBM Class:   F            PrivClasses: F
  CMDBK Address: 001CBBE8   Entry Point: HCPRSCDI
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid command name - *command*
- HCP6706E Invalid IBM class - *c*
- HCP6706E Invalid privilege classes - *classes*
- HCP6706E Invalid string - *string*
- HCP6706E Invalid subcommand name - *subcommand*
- HCP6720I No items found to satisfy selection criteria.
- HCP6840E You specified QUERY VIRTUAL without SUBCMD

## QUERY CPDISKS



### Authorization

Privilege Class: A

### Purpose

Use QUERY CPDISKS to display the status of a CP-accessed minidisk or all CP-accessed minidisks.

### Operands

**\***

**fm**

is the file mode letter of the CP-accessed minidisk. If omitted or specified as an asterisk (\*), the default, CP displays the status of all the disks that it has accessed.

### Usage Notes

1. To have CP access a minidisk, you can use the CPACCESS command, or you can specify a CP\_ACCESS statement in the system configuration file.

For more information on the CPACCESS command, see [“CPACCESS” on page 89](#). For more information on the system configuration file, see [CP\\_ACCESS Statement in z/VM: CP Planning and Administration](#).

### Responses

Response 1:

If you enter QUERY CPDISKS with no additional parameters, you see something like this:

Label	Userid	Vdev	Mode	Stat	Vol-ID	Rdev	Type	StartLoc	EndLoc
CF1	MAINT	0CF1	A	R/O	ESARES	0A0F	CKD	2400	2449
CFG1	MAINT	0300	B	R/O	ESARES	0A0F	CKD	95	99
CFBA1	MAINT	0580	F	R/O	ESAFBA	0B40	FBA	45490	45839

Ready;

In this example,

#### Label

is the minidisk label that was assigned to the disk when it was formatted.

#### Userid

is the user ID of the minidisk owner.

#### Vdev

is the virtual device number of the minidisk, as specified in the owner's directory entry in the system directory.

#### Mode

is the file mode that CP used to access the minidisk.

#### Stat

shows whether the minidisk is in read-only (R/O) or read-write (R/W) mode.



**Vol-ID**

is the real volume serial of the DASD volume that contains the minidisk.

**Rdev**

is the real device number of the DASD volume that contains the minidisk.

**Type**

is the device type: CKD or FBA.

**StartLoc**

is the starting cylinder or block of the minidisk.

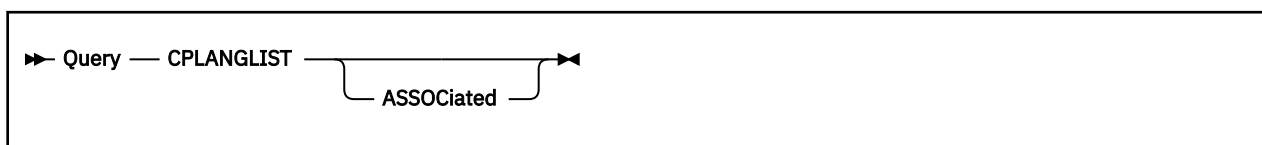
**EndLoc**

is the ending cylinder or block of the minidisk.

**Messages**

- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP6706E Invalid disk mode - *filemode*
- HCP6727I Disk *filemode* not accessed
- HCP6727I No disks accessed

## QUERY CPLANGLIST



### Authorization

Privilege Class: G

### Purpose

Use QUERY CPLANGLIST to display all the valid:

- Languages in which CP can display control program messages to your virtual machine, or
- Message repository languages available on the system, their component names, and their external symbols, if any.

### Operands

#### ASSOCIATED

tells CP to display all the valid message repository languages available on the system, their component names, and their external symbols, if any.

If omitted, CP displays all the valid languages in which CP can display control program messages to your virtual machine.

### Usage Notes

1. When you specify the ASSOCIATED operand, CP displays message repository information: language identifiers, component identifiers, and external symbol names. This information was associated with the message repositories using the ASSOCIATE MESSAGES or MSGS command (see [“ASSOCIATE MESSAGES / MSGS”](#) on page 24).
2. For more information about user-defined message repositories, see [Understanding the CP Message Repository](#) in *z/VM: CP Exit Customization*.
3. The system default language is the language that is specified in the system configuration file, or the language that was specified by the operator using the SYSTEM operand of the SET CPLANGUAGE command. If neither of these was specified, then the language is the one built into the CP nucleus.
4. If you change the system default language, the response to QUERY CPLANGLIST shows the new system default language. If the user is using the system default language, the user's current language will not change until the user either logs off and logs back on, or issues the CP SET CPLANGUAGE command or CMS SET LANGUAGE command to switch to the new system default language.

### Responses

Response 1:

```

System default language:      xxxxx
User's current language:     xxxxx
Other available languages:   xxxxx
:
```

#### System default language

is the default language for CP.

**User's current language**

is the language that is currently active for CP in your virtual machine.

**Other available languages**

are other available CP languages (or NONE).

**xxxxx**

is the abbreviation, up to five characters, for a language that is valid in your virtual machine.

Response 2:

To display the message repository languages, their component IDs, and their external symbol names, enter the following:

```
query cplanglist associated
Language: AMENG Component: AC$ EPName: AC$CSCT1
Language: UCENG Component: AC$ EPName: AC$CSCT2
Ready;
```

This response tells you that you have 2 local message repositories that someone on your system added with CPXLOAD and ASSOCIATE MESSAGES or MSGS commands. The first one is in American English (AMENG) and the second is in uppercase English (UCENG).

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*

## QUERY CPLANGUAGE

---

► Query — CPLANGUage ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY CPLANGUAGE to identify the language being used to display control program messages to your virtual machine.

### Responses

#### Response 1:

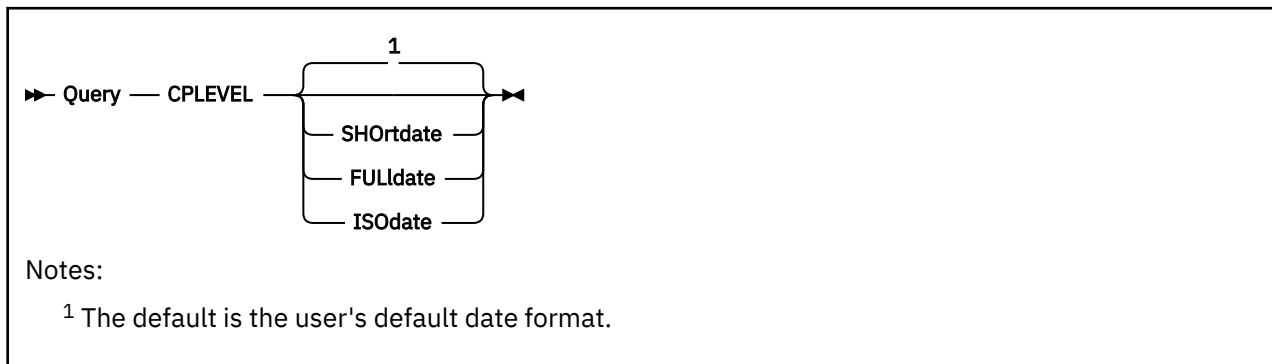
*language*

identifies the CP language currently being used by the virtual machine.

### Messages

- HCP002E Invalid operand - *operand*

## QUERY CPLEVEL



### Authorization

Privilege Class: G

### Purpose

Use QUERY CPLEVEL to determine the following control program attributes:

- The software version level, release level, and release modification level
- The software service level number
- The date and time (translated to the current active time zone) that the CP system software was written to DASD.
- The date and time the CP was last started

### Operands

#### SHOrtdate

specifies that dates in the response to this command will be displayed in *mm/dd/yy* format, where *mm* is the month, *dd* is the day of the month, and *yy* is the 2-digit year.

#### FULldate

specifies that dates in the response to this command will be displayed in *mm/dd/yyyy* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

#### ISOdate

specifies that dates in the response to this command will be displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

### Usage Notes

1. The default date format for certain CP and CMS commands can be set on a system-wide basis and also for the individual user. The system-wide default date format is set with the SYSTEM\_DATEFORMAT system configuration statement. The user's default date format is set with the DATEFORMAT directory statement. The system-wide default and the user's default can also be set with the CP SET DATEFORMAT command. The user's default date format defaults to the system-wide default. The system-wide and user settings can be queried with the CP QUERY DATEFORMAT command. The hierarchy of possible date format settings for the QUERY CPLEVEL command, from highest priority to lowest, is:
  - QUERY CPLEVEL command operand
  - User default
  - System-wide default

- The output of the QUERY CPLEVEL command always reflects the level of CP code on which this virtual machine is running. This is true even when the virtual machine's virtual architecture limits its use of CP-provided features to a lower level of CP functionality. This occurs when the virtual machine is assigned to a relocation domain that includes one or more members running at a lower level of CP than the one on which this virtual machine is running. Therefore the output of this command should not be used to determine the level of CP-provided functionality available to the virtual machine. Diagnose X'00' is the proper interface to determine the virtual machine's virtual CP level. For an explanation of relocation domains, see [Using Relocation Domains in z/VM: CP Planning and Administration](#) for more details. See the documentation for [“QUERY VMRELOCATE” on page 1348](#) and [“SET VMRELOCATE” on page 1755](#), and the [VMRELOCATE Directory Statement](#) for details about how a virtual machine is assigned to a relocation domain.

## Responses

Response 1:

If you enter QUERY CPLEVEL SHORTDATE, you get the following response:

```
z/VM VERSION v RELEASE r.m, SERVICE LEVEL yyyn (64-BIT)
GENERATED AT mm/dd/yy hh:mm:ss timezone
IPL AT mm/dd/yy hh:mm:ss timezone
```

Response 2:

If you enter QUERY CPLEVEL FULLDATE, you get the following response:

```
z/VM VERSION v RELEASE r.m, SERVICE LEVEL yyyn (64-BIT)
GENERATED AT mm/dd/yyyy hh:mm:ss timezone
IPL AT mm/dd/yyyy hh:mm:ss timezone
```

Response 3:

If you enter QUERY CPLEVEL ISODATE, you get the following response:

```
z/VM VERSION v RELEASE r.m, SERVICE LEVEL yyyn (64-BIT)
GENERATED AT yyyy-mm-dd hh:mm:ss timezone
IPL AT yyyy-mm-dd hh:mm:ss timezone
```

**v**  
identifies the software version level.

**r.m**  
identifies the software release level and the release modification level.

### **SERVICE LEVEL yyyn**

identifies the software service level number. The number indicates the most recent RSU service tape that has been applied. *yy* is the last 2 digits of the year and *nn* is the sequential number of the RSU tape for that year. It cannot indicate which individual updates have been incorporated into CP. The system programmer can find out what individual updates have been incorporated by using the VMSES/E tool. For more information, see [Using VMSES/E for Service](#) in *z/VM: VMSES/E Introduction and Reference*.

### **GENERATED AT mm/dd/yy hh:mm:ss timezone**

### **GENERATED AT mm/dd/yyyy hh:mm:ss timezone**

### **GENERATED AT yyyy-mm-dd hh:mm:ss timezone**

indicates the date and time (translated to the current active time zone) that the CP system software was written to DASD. One of the above responses is generated depending on the date format specified on the user's default date format.

### **IPL mm/dd/yy hh:mm:ss timezone**

### **IPL mm/dd/yyyy hh:mm:ss timezone**

### **IPL yyyy-mm-dd hh:mm:ss timezone**

indicates the date and time the CP system software was last started. One of the above responses is generated depending on the date format specified on the user's default date format.

**Messages**

- HCP003E Invalid option - command contains extra option(s) starting with *operand*

## QUERY CPLOAD

► Query — CPLOAD ◄

### Authorization

Privilege Class: A, B, E

### Purpose

Use QUERY CPLOAD to display information regarding the last CP IPL. The information displayed includes the location of the CP module that was last used, the location of the parm disk, and how CP was started.

You can load CP only from a module on a CMS-formatted minidisk. When you load CP, you must specify device, system, and logo definition information. You specify this information in files on a parm disk. CP can 'sense' most device definition information, so you do not have to define it. System definition information goes in the system configuration file, and logo definition information goes in the logo configuration file.

### Responses

**Note:** When you use FBA devices, the responses are shown in blocks.

Response 1:

For CKD/ECKD devices:

```
Module CPLOAD2 was loaded from minidisk on volume ESARES at cylinder 2400
Parm disk on volume ESARES, cylinders 2400 through 2420
Last start was system IPL.
```

For FBA devices:

```
Module CPLOAD2 was loaded from minidisk on volume FBARES at block 24000
Parm disk on volume FBARES, blocks 24000 through 43992
Last start was a system IPL.
```

The first line of the response tells you CP was loaded from a module, the name of the module, and the location of the minidisk from which CP loaded the module. The second line of the response tells you CP used a parm disk during the IPL and the location of the parm disk.

CP assumes the parm disk is the same minidisk that contains the module, unless you override that assumption. In the example above, the second line of the response indicates that the person who performed the IPL either:

- Did not override the system's assumption. The Stand Alone Loader used a cylinder or block offset to find the location of the parm disk. The allocation map did not have a parm disk defined at that location.
- Used the PDOFF IPL parameter during the IPL to specify a cylinder or block offset of the location of the parm disk. The value specified for the PDOFF IPL parameter coincides with the location of the minidisk from which CP loaded the module. The allocation map did not have a parm disk defined at that location.

For more information on the PDOFF parameter or any of the other IPL parameters, see [Using the Stand-Alone Program Loader in z/VM: System Operation](#).

To display the current IPL parameters, use the CP QUERY IPLPARMS command. Be aware that the CP SET IPLPARMS command may have changed them since CP was started. For more information on these commands, see [“QUERY IPLPARMS” on page 954](#) and [“SET IPLPARMS” on page 1530](#).

The third line of the response tells you that CP was started by the system operator at the system console. One of the following variations will appear as the third line of the response.



- The response below tells you that CP was started by the system operator at the system console.

```
Last start was a system IPL.
```

The response below tells you that the operator restarted CP using the CP SHUTDOWN command with the REIPL operand.

```
Last start was a system restart from SHUTDOWN REIPL.
```

- The response below tells you that CP restarted itself after abend XXXnnn occurred. Use CP QUERY CPLEVEL to find out when the restart occurred.

```
Last start was a system restart after system failure XXXnnn
```

#### Response 2:

```
Module CPLOAD was loaded from minidisk on volume ESARES at cylinder 20
Parm disk number 2 on volume ESARES, cylinders 90 through 94
Last start was a system IPL
```

In this example, the second line of the response indicates that the person who performed the IPL either:

- Did not override the system's assumption and the allocation map had a parm disk defined for the minidisk containing the CP module.
- Or, used the PDNUM parameter during the IPL to specify a specific parm disk number, as defined in the allocation map
- Or, used the PDOFF parameter during the IPL to specify a cylinder or block offset and the allocation map had a parm disk defined in that location

For more information on the PDNUM or PDOFF parameter or any of the other IPL parameters, see [Using the Stand-Alone Program Loader](#) in *z/VM: System Operation*.

#### Response 3:

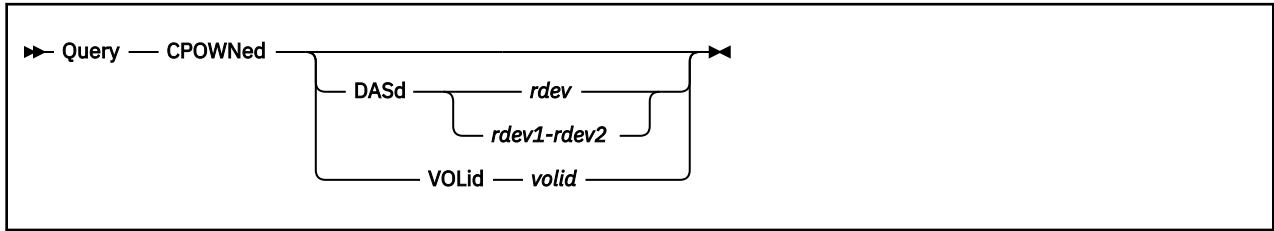
```
Module NEWCP loaded from minidisk on volume ESARES at cylinder 2400
Parm disk number 3 on volume OLDRES, cylinders 586 through 590
Last start was a system IPL.
```

In this example, the volume IDs on the first and second lines are different. The person who performed the IPL loaded CP from a module on one volume and used a parm disk on another volume.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*

## QUERY CPOWNED



### Authorization

Privilege Class: G

### Purpose

Use QUERY CPOWNED to display the list of CP-owned DASD volumes.

### Operands

**DASd rdev**

**DASd rdev1-rdev2**

tells CP to display information on only the CP-owned volumes whose addresses are in the range specified.

**VOLid valid**

tells CP to display information on only the *valid* specified. You can specify the *valid* as a generic volume identifier. A generic volume identifier is a 1- to 6-character pattern that can use asterisks (\*) in place of zero or more characters and percents (%) in place of exactly one character. For example, VOLID \*EM% displays all CP-owned volumes ending with a 3-character sequence that begins with the letters 'EM'.

### Usage Notes

1. If you specify no parameters on the QUERY CPOWNED command, CP displays information on all CP-owned volumes, as well as gaps in the volume list.
2. You can use the DEFINE CPOWNED command (see “DEFINE CPOWNED” on page 150) to bring new CP-owned volumes on line into empty slots in the volume list. CP allocates empty slots for every slot number that was skipped in the system configuration file. You can define additional empty slots at the end of the list of CP-owned DASD by explicitly defining them in the system configuration file with CP\_OWNED statements.

### Responses

Response 1:

If you enter the QUERY CPOWNED command with no operands on system GDLSYSME in SSI cluster GDLSI, you might see a response similar to this:

Slot	Vol-ID	Rdev	Type	Status	SSIOwner	SysOwner
1	ESARES	0A0F	Own	Online and attached	GDLSI	GDLSYSME
2	ESAP01	0A01	Own	Online and attached	GDLSI	GDLSYSME
3	ESAP02	0A02	Share	Online and attached	GDLSI	GDLSYS1
4	ESAP03	0A03	Dump	Online and attached	GDLSI	GDLSYSME
5	-----	----	---	Reserved	-----	-----
6	ESAP05	0A05	Share	Online and attached	GDLSI	GDLSYS1
7	ESAP06	0A06	Share	Online and attached	GDLSI	GDLSYS1
8	ESAP07	0A07	Share	Online and not attached	GDLSI	GDLSYS1

9	ESAP08	0A08	Own	Online and attached	GDLSSI	GDLSYSME
10	ESAP09	----	Own	Offline	-----	-----
11	ESAP0A	0A0A	Own	Online and attached	GDLSSI	GDLSYSME
12	ESAP0B	0A0B	Share	Online and attached	GDLSSI	GDLSYS1
13	ESAP0C	0A0C	Share	Online and attached	GDLSSI	GDLSYS1
14	ESAP0D	0A0D	Share	Online and attached	GDLSSI	GDLSYS1

If you enter the QUERY CPOWNER command with no operands on system GDLSYSME in a non-SSI environment, using the same configuration as in the previous example, you might see a response similar to this:

Slot	Vol-ID	Rdev	Type	Status	SSIOwner	SysOwner
1	ESAP08	0A08	Own	Online and attached	-----	GDLSYSME
2	ESAP01	0A01	Own	Online and attached	-----	-----
3	ESAP02	0A02	Own	Online and attached	-----	-----
4	ESAP03	0A03	Dump	Online and attached	-----	GDLSYSME
5	-----	----	---	Reserved	-----	-----
6	ESAP05	0A05	Own	Online and attached	-----	-----
7	ESAP06	0A06	Own	Online and attached	-----	-----
8	ESAP07	0A07	Own	Online and not attached	-----	-----
9	ESAP08	0A08	Own	Online and attached	-----	GDLSYSME
10	ESAP09	----	Own	Offline	-----	-----
11	ESAP0A	0A0A	Own	Online and attached	-----	-----
12	ESAP0B	0A0B	Own	Online and attached	-----	-----
13	ESAP0C	0A0C	Own	Online and attached	-----	-----
14	ESAP0D	0A0D	Own	Online and attached	-----	-----

The SSI example shows that the volumes are marked with ownership information. The non-SSI example shows that volumes can be marked with ownership information, although this is not required.

In both examples, slot 5 is reserved, and you can use the DEFINE CPOWNER command to add a new volume to the list.

**Own**

indicates that the volume is owned by the local system, which is able to read and write SPOOL files.

**Share**

indicates that the volume is owned by another system in the SSI cluster. The local system can only read SPOOL files that were previously written by the owning system. Note that SHARED can be displayed on a non-SSI system if the volume attached is **not** owned by this system.

**Dump**

indicates that the spool space on the volume is reserved exclusively for dumps.

**Reserved**

indicates that a volume is not associated with the slot, and the slot is available for allocation.

**Offline**

indicates that a volume with this label does not exist, or one does exist but is not attached to the system.

**Online and attached**

indicates that the volume exists, is online, and is attached to the system. This is the normal state of CP-owned volumes.

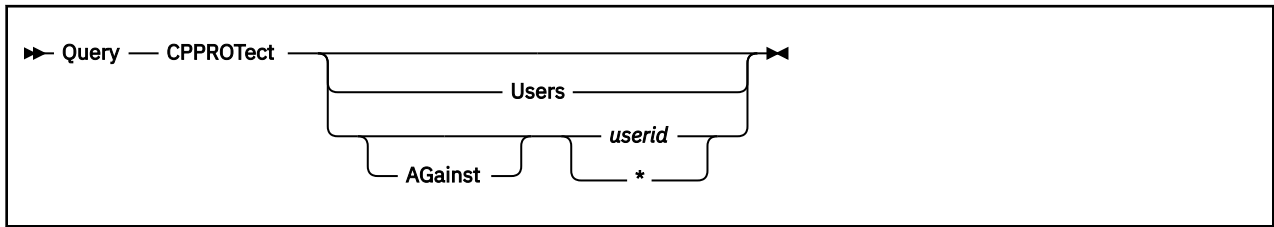
**Online and not attached**

indicates that the volume exists, is online, but is not yet attached to the system. This is a transient case and not likely to be displayed.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP6704E Missing token at end of line
- HCP6706E Invalid {address range|volume label} - {*range|valid*}
- HCP6726E Could not find any CP-owned volumes that met the specified criteria.

## QUERY CPPROTECT



### Authorization

Privilege Class: C

### Purpose

Use QUERY CPPROTECT to display protection settings of the system, of users with protection overrides, or of a specific user, as well as availability of the required machine support for each protection mode.

In addition, if the machine support was not present when z/VM was IPLed and the required service is dynamically applied later, QUERY CPPROTECT can be issued to cause CP to recognize the machine support and put previously requested CPPROTECT settings into effect.

### Operands

#### Users

includes a display of the MODE2 settings for all users for whom a MODE2 override has been requested.

#### AGainst *userid*

displays only the MODE2 settings for a specific user.

#### AGainst \*

displays only the MODE2 settings for the command issuer.

### Usage Notes

1. MODE1 refers to a system-wide mitigation mechanism. MODE2 refers to a mitigation mechanism that the z/VM Control Program (CP) can apply when it interacts with some users but not with others. Details on the protection modes are not provided in this publication. Authorized personnel of IBM clients can consult the IBM Z Security Portal for further information and guidance. To request access to the portal, see the instructions referenced in the SET CPPROTECT command description.
2. To remove a user override, issue: `set cprotect mode2 against userid default.`
3. To remove all user overrides, issue: `set cprotect mode2 against all default.`
4. Users who do not appear in the QUERY CPPROTECT USERS output are operating with the default MODE2 setting.

### Examples

Response 1:

The response to QUERY CPPROTECT consists of the following three lines.

Line 1:

```
MODE1 {AVAILABLE } REQUESTED {ON } CURRENTLY {ON }
      {UNAVAILABLE } {OFF } {OFF }
```

This line indicates whether the machine support required for MODE1 function is present in the machine on which z/VM is running. The REQUESTED clause shows what setting was requested. The CURRENTLY clause shows how the system is currently running. These will differ only if the machine support is absent.

Line 2:

```
MODE2 {AVAILABLE } REQUESTED {ON } CURRENTLY {ON }
      {UNAVAILABLE }          {OFF }          {OFF }
      {INHERENT }             {SELECTIVE } {SELECTIVE }
```

This line indicates whether the machine support required for MODE2 function is present in the machine on which z/VM is running, and if present, whether it is available for optional activation or provided inherently in the machine design. The REQUESTED and CURRENTLY clauses summarize the settings that were requested and those currently in effect. These will differ if ON was requested but the machine support is absent, or if OFF was requested but the machine design inherently delivers the protection. ON and OFF indicate a setting that was requested or is currently active as the system default, with no user overrides in place. SELECTIVE indicates that there are one or more user overrides requested and currently in effect, respectively. Details on the overrides can be displayed using QUERY CPPROTECT USERS.

Line 3:

```
MODE2  DEFAULT          REQUESTED {ON } CURRENTLY {ON }
                          {OFF }          {OFF }
```

This line indicates the system-wide default setting for MODE2 as set by the most recent SET CPPROTECT MODE2 DEFAULT command. If no such command has been issued, the DEFAULT OFF setting established at z/VM IPL is shown. The default setting applies to all users except those for whom an override has been set via SET CPPROTECT MODE2 AGAINST *userid*.

Response 2:

The response to QUERY CPPROTECT USERS consists of Response 1 above, followed by zero or more instances of line 4, as shown below.

Lines 4 to n:

```
MODE2  AGAINST userid  REQUESTED {ON } CURRENTLY {ON }
                          {OFF }          {OFF }
```

Line 4 in this response shows the override setting for a user, as set by SET CPPROTECT MODE2 AGAINST *userid* {ON|OFF}. The override value applies to that user regardless of the system default or any changes to the system default. As above, the REQUESTED and CURRENTLY clauses indicate the saved request and the actual setting, according to the presence of the machine support for MODE2.

Response 3:

The response for QUERY CPPROTECT AGAINST *userid* consists of one line indicating the requested user's MODE2 setting, as shown below.

```
MODE2  AGAINST userid  REQUESTED {ON } CURRENTLY {ON }
                          {OFF }          {OFF }
                          {DEFAULT }
```

This response is displayed for the specified user, regardless of whether an override has been established for the user. The REQUESTED clause indicates whether the user is configured to operate according to an ON or OFF per-user override or according to the system-wide default setting. The CURRENTLY clause indicates how protection is being applied to the user, based on the REQUESTED response, the system-wide default, and the presence of the required machine support.

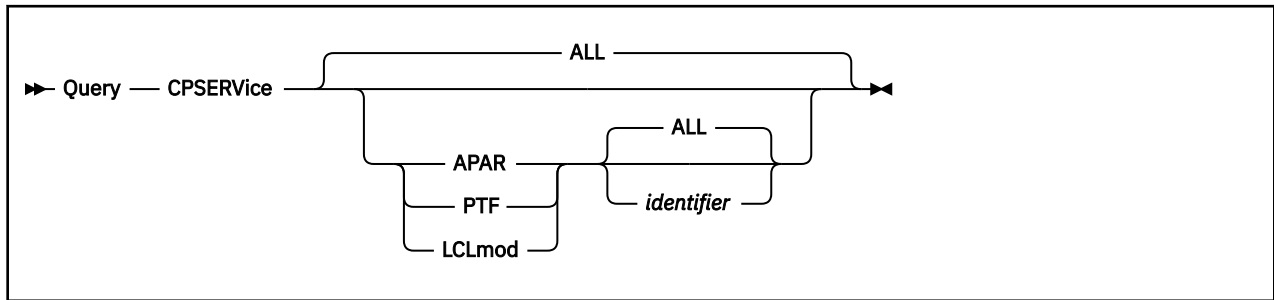
## Messages

- HCP002E Invalid operand - *operand*

## QUERY CPPROTECT

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on

## QUERY CPSERVICE



### Authorization

Privilege Class: C, E

### Purpose

Use QUERY CPSERVICE to display information about the APARs, PTFs and local modifications applied to the currently running CP module.

### Operands

#### ALL

displays all APARs, their corresponding PTFs, and all local modifications applied to the system.

#### APAR *identifier*

displays the specified APAR and its corresponding PTF, if it is applied to the system. A message is displayed if the specified APAR is not applied to the system. If no APAR number is specified, or if the ALL keyword is used instead of an APAR number, all APARs applied to the system and their corresponding PTFs are displayed. An asterisk (\*) can be used as a wildcard character to represent one or more characters at the end of an identifier string. The identifier should be 1-8 characters long.

#### LCLmod *identifier*

displays the specified local modification, if it is applied to the system. A message is displayed if the specified local modification is not applied to the system. If no local modification name is specified, or if the ALL keyword is used instead of a local modification name, all local modifications applied to the system are displayed. An asterisk (\*) can be used as a wildcard character to represent one or more characters at the end of an identifier string. The identifier should be 1-8 characters long.

#### PTF *identifier*

displays the specified PTF and its corresponding APAR, if it is applied to the system. A message is displayed if the specified PTF is not applied to the system. If no PTF number is specified, or if the ALL keyword is used instead of an PTF number, all PTFs applied to the system and their corresponding APARs are displayed. An asterisk (\*) can be used as a wildcard character to represent one or more characters at the end of an identifier string. The identifier should be 1-8 characters long.

### Responses

Response 1:

This response is displayed if an APAR or PTF is available on the system, where *aparnum* and *ptfnum* are 1- to 8-character strings that represent APAR and PTF numbers, respectively.

APAR	PTF
<i>aparnum</i>	<i>ptfnum</i>

## QUERY CPSERVICE

For example, QUERY CPSERVICE APAR ALL displays results similar to the following:

```
APAR      PTF
VMRSU02   UMRSU02
VMRSU04   UMRSU04
VM65348   UM34005
VM65352   UM34010
```

QUERY CPSERVICE PTF UM3404\* displays results similar to the following:

```
APAR      PTF
VM65371   UM34046
VM65373   UM34042
VM65375   UM34044
```

Response 2:

This response is displayed if a local modification is available on the system, where *localmod* is a one- to eight-character string that represent a local modification identifier.

```
Local modification
Localmod
```

For example, QUERY CPSERVICE LCLMOD ALL displays results similar to the following:

```
Local modification
PVGDPBYE
PVGD2BXC
SQ475BY0
```

QUERY CPSERVICE LCLMOD MY\* displays results similar to the following:

```
Local modification
MYLCLMOD
MYOTRMOD
```

Note that the service type is sorted in alphanumeric order. If QUERY CPSERVICE ALL is specified, then the APARs will appear first, followed by local modifications, with a single blank line between sections:

```
APAR      PTF
VMRSU02   UMRSU02
VMRSU04   UMRSU04
VM65348   UM34005
VM65352   UM34010

Local modification
PVGDPBYE
PVGD2BXC
SQ475BY
```

Response 3:

This response is displayed when the currently running CP module does not contain any information about CP service that is applied to the system.

```
No APAR/PTF data found.
```

This response is displayed for either of the following reasons:

- No CP APARs are applied to the system
- The SERVICE command was not able to include CP service information in the CP module when it was built. If you believe that this is the case, you should review the service message log (\$VMFSRV \$MSGLOG) for any warnings or errors related to including service information in the CP module.

Response 4:

This response is displayed when the currently running CP module does not contain any information about local modifications that are applied to the system:



```
No local modification data found.
```

This response is displayed for either of the following reasons:

- No local modifications are applied to the system
- The SERVICE command was not able to include local modification information in the CP module when it was built. If you believe that this is the case, you should review the service message log (\$VMFSRV \$MSGLOG) for any warnings or errors related to including service information in the CP module.

Response 5:

This response is displayed when no information about CP service or CP local modifications is available in the currently running CP module:

```
No service content data is available.
```

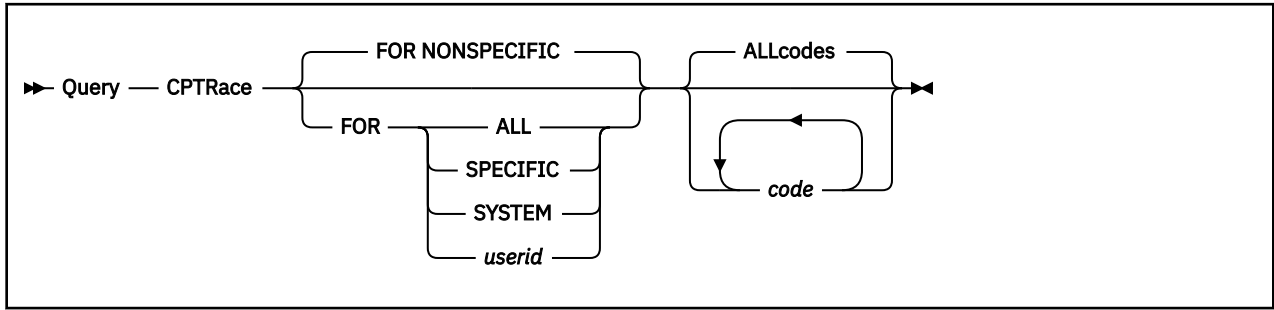
This response is displayed for either of the following reasons:

- No CP service or local modifications are applied to the system
- The SERVICE command was not able to include CP service and/or local modification information in the CP module when it was built. If you believe that this is the case, you should review the service message log (\$VMFSRV \$MSGLOG) for any warnings or errors related to including service information in the CP module.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP2119E The CP service data included in the CP nucleus is incompatible with the QUERY CPSERVICE command.
- HCP6720I No items found to satisfy selection criteria.

## QUERY CPTRACE



### Authorization

Privilege Class: A, C, E

### Purpose

Use QUERY CPTRACE to display status of the trace function for real machine events.

**Note:** This status is specified by SET CPTRACE.

### Operands

#### **NONSPECIFIC**

#### **SPECIFIC**

#### **SYSTEM**

#### **ALL**

#### *userid*

NONSPECIFIC indicates that the specified tracing applies to all virtual machines not specifically set or named by a previous SET CPTRACE command. NONSPECIFIC is the default. All virtual machines begin in the NONSPECIFIC group. All members of the NONSPECIFIC group have the same codes traced.

SPECIFIC indicates that the specified tracing applies only to those virtual machines specifically set or named by a previous SET CPTRACE command. Members of the SPECIFIC group may each have a unique set of codes traced.

SYSTEM indicates that the specified tracing applies to system work. SYSTEM becomes a member of the SPECIFIC group even though the specified tracing may not match the virtual machines already in the group. Therefore, SYSTEM is similar to *userid* in that it names the work for which tracing applies, and SYSTEM can also be a part of either the SPECIFIC or NONSPECIFIC groups.

ALL indicates that the specified tracing applies to both the SPECIFIC and NONSPECIFIC groups of virtual machines and SYSTEM.

The operand *userid* indicates that the specified tracing applies only to this specific virtual machine; *userid* becomes a member of the SPECIFIC group, even though the specified tracing may not match others in the group.

#### **ALLcodes**

displays the setting of all CPTRACE categories.

#### **code**

identifies a single four-digit trace code or the name of a category of codes. Only the setting of this code or category of codes is displayed.

## Usage Notes

1. The SET CPTRACE command determines which events are recorded. QUERY CPTRACE lets you know what tracing events have been set on or off.

To check for *only* NONSPECIFIC categories, enter `query cptrace`.

To check for *only* SPECIFIC categories, enter `query cptrace for specific`.

2. You can use the QUERY CPTRACE response to recreate the CPTRACE setting at some later point. For each response line, in the order displayed, the indicated code or category is set to ON if ENABLED or ENABLED EXCEPT, and is set to OFF if DISABLED or DISABLED EXCEPT. In some cases (see Example 4) the CPTRACE setting can be recreated by entering the following:

```
set cptrace for nonspecific off allcodes
set cptrace for nonspecific on svc
set cptrace for nonspecific off 0200
set cptrace for nonspecific off 3600
set cptrace for nonspecific on realios
set cptrace for nonspecific off ssch
```

## Responses

Response 1:

The QUERY CPTRACE response shows an *indented hierarchy of trace categories* describing, as concisely as possible, the current CPTRACE setting. The output takes the form:

```
FOR qualifier
category ENABLED/DISABLED EXCEPT
  subcategory DISABLED/ENABLED EXCEPT
    subcategory ENABLED/DISABLED
      subcategory DISABLED/ENABLED
```

Response 2:

For categories not specifically traced, the output is:

```
FOR qualifier NO SPECIFIC TRACE SET
```

Response 3:

For a query on the SPECIFIC qualifier, when no specific tracing is in effect, the output is NO SPECIFIC TRACE SET. This response is repeated for each qualifier for which the trace setting is to be displayed. In the following examples, no specific tracing is assumed unless otherwise indicated.

**Note:** For further information on trace categories, see [“SET CPTRACE” on page 1446](#).

## Examples

Example 1:

If all TRACE entries were enabled through the command SET CPTRACE ON (or all TRACE entries were disabled through the command SET CPTRACE OFF), you receive the following response:

```
FOR NONSPECIFIC
ALLCODES ENABLED or (DISABLED)
```

Example 2:

All TRACE entries were enabled through the command SET CPTRACE ON. If you wanted to display the setting of the REALIOS trace category after deactivating trace for CLEAR SUBCHANNEL (X'1000' and

## QUERY CPTRACE

X'1003') through the command SET CPTRACE OFF CSCH, you would enter `query cptrace realios`. You would receive the following response:

```
FOR NONSPECIFIC
REALIOS  ENABLED EXCEPT
CSCH     DISABLED
```

Example 3:

If all TRACE entries were enabled through the command SET CPTRACE ON and you wanted to disable the TRACE entries for CALL-WITH-SAVE AREA (X'2800'), RETURN-WITH-SAVE AREA (X'2C00'), RUN USER (X'0A00'), and IUCV (X'1500'- X'1511') through the command `set cptrace off callret 0A00 iucv`, you receive the following response:

```
FOR NONSPECIFIC
ALLCODES ENABLED EXCEPT
CALLRET  DISABLED
0A00     DISABLED
IUCV     DISABLED
```

Example 4:

All TRACE entries were disabled through the command SET CPTRACE OFF. Perhaps you want to enable all the TRACE entries in category SVC and REALIOS except for SVC INTERRUPTION (X'0200'), EXIT TO THE DISPATCHER (X'3600'), and START SUBCHANNEL (X'1030'-'1033'). To do so, enter `set cptrace on svc realios off 0200 3600 ssch`. You receive the following response:

```
FOR NONSPECIFIC
ALLCODES DISABLED EXCEPT
SVC      ENABLED EXCEPT
0200     DISABLED
3600     DISABLED
REALIOS  ENABLED EXCEPT
SSCH     DISABLED
```

Example 5:

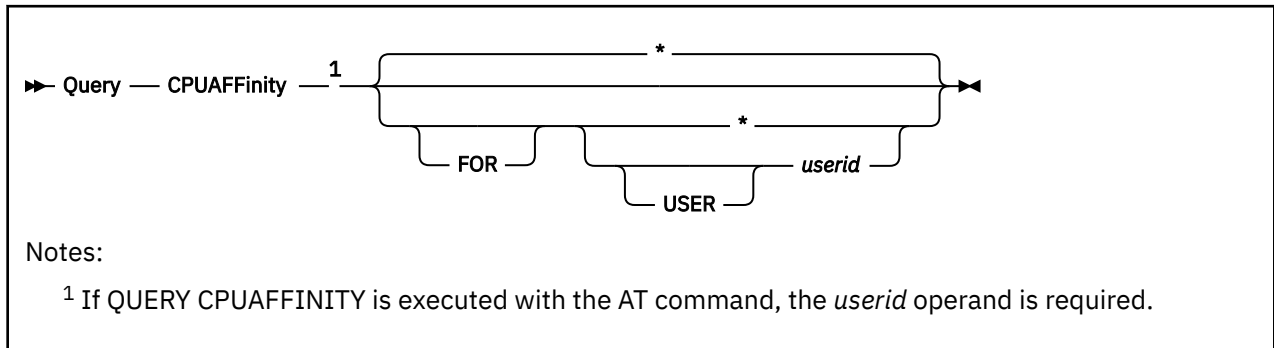
All tracing is enabled for the nonspecific and disabled for BATCH1 by issuing `set cptrace on` and `set cptrace for batch1 off`. You receive the following response for QUERY CPTRACE for all codes:

```
FOR NONSPECIFIC
ALLCODES ENABLED
FOR BATCH1
ALLCODES DISABLED
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP053E [XAUTOLOG failed for *userid*:] {*userid*|*value*} not in CP directory
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]

## QUERY CPUAFFINITY



### Authorization

Privilege Class: A

### Purpose

Use QUERY CPUAFFINITY to display the setting for CPU affinity.

### Operands

\*

displays the CPU affinity setting for the user issuing the command.

#### USER *userid*

specifies the user ID of the specific virtual machine for which you want to display the CPU affinity setting.

### Usage Notes

1. A class G user can use QUERY VIRTUAL CPUS to determine the user's own CPU affinity setting.
2. ON means the virtual CPUs are dispatched on only real CPUs of matching type. OFF means the virtual CPUs are dispatched on only real CPUs of primary type.

### Responses

Response 1:

```
CPUAFFINITY = ON
              OFF
```

### Messages

- HCP002E Invalid operand *operand*
- HCP003E Invalid option - *{option | command contains extra option(s) starting with option}*
- HCP045E *userid* not logged on
- HCP6704E Missing token at end of line

## QUERY CPUID

► Query — CPUid ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY CPUID to display the doubleword processor identifier used by your virtual machine.

### Usage Notes

1. This command returns the CPUID of the issuing user's VIRTUAL processor. The virtual CPUID defaults to the real configuration master processor CPUID unless the user has issued SET CPUID or is using the directory OPTIONS or CPU statements to alter the virtual CPUID, or the user is in a relocation domain that requires its architecture to appear at a lower level than the real configuration master processor.
2. To obtain the CPUID of non-base processors in a virtual configuration, use the CPU command to direct the Q CPUID to each non-base processor.
3. When part of an SSI cluster, the results of QUERY CPUID depend on the relocation domain of the issuing user. This command shows the virtual CPUID of the issuing user. Virtual CPUIDs may differ among users on the same system when they belong to different relocation domains.

### Responses

Response 1:

```
CPUID= aa s s s s s s c c c c d d d d
```

#### **aa**

identifies the version code. These two digits are set to X'FF' to identify that your virtual machine is running under z/VM.

#### **ssssss**

identifies the processor. This field contains six hexadecimal digits and is set at logon from the real configuration master processor. This is the only part of the CPUID that can be modified by means of the SET CPUID command or set by the system directory's OPTION control statement. This portion of the CPUID is not modified if the owning guest is relocated.

#### **cccc**

identifies the model number. This field is set to the model number of the real machine unless the user is in a relocation domain that requires its architecture to appear at a lower level than the real configuration master processor, in which case it is set to the level required by the relocation domain.

#### **dddd**

is bits 48-63 of the CPU identification of the real machine. On all supported servers, this field is set to X'8000'. For details on the value reported by QUERY CPUID, see the STORE CPU ID control instruction in [z/Architecture Principles of Operation](#).

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

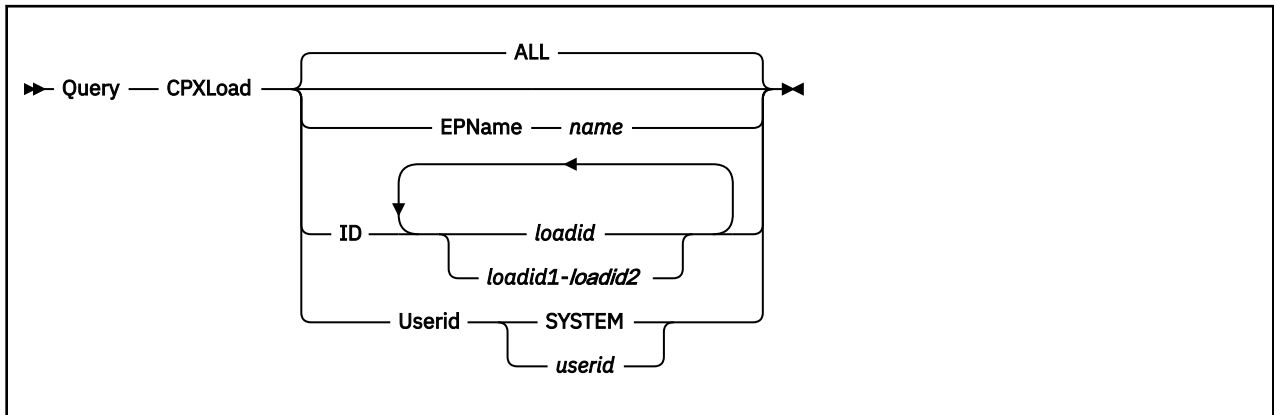
## QUERY CPUPOOL

---

### **Purpose**

QUERY CPUPOOL has been superseded by QUERY RESPOOL. See [“QUERY RESPOOL”](#) on page 1140.

## QUERY CPXLOAD



### Authorization

Privilege Class: A, C, E

### Purpose

Use `QUERY CPXLOAD` to display information about the customer-written CP routines that were loaded using the `CPXLOAD` command or configuration file statement and that currently exist in the system execution space.

### Operands

#### ALL

tells CP that you want information about all of the customer-written CP routines that were loaded into the system using `CPXLOAD` commands or configuration file statements and that have not been removed from the system with `CPXUNLOAD` commands. If you do not specify any operands, the default is `ALL`.

#### EPName *name*

tells CP that you only want information about the customer-written CP routines associated with the specified entry point name or names. The variable *name* must be a 1-character to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

You can use generic entry point names to request information about a specific subset of entry point names. A generic entry point name is a 1-character to 8-character string with asterisks (\*) in place of 1 or more characters and percent signs (%) in place of exactly 1 character. For example,

```
query cpxload epname hcp%rc*
```

displays information about all the entry points that start with HCP and have RC as their fifth and sixth characters.

#### ID *loadid*

#### ID *loadid1-loadid2*

is the number of the load identifier associated with the customer-written CP routines. CP assigned this load ID when you used the `CPXLOAD` command or configuration file statement to load these CP routines into the system execution space. Each *loadid* must be a decimal number between 0 and 2147483647.



**Userid SYSTEM**

tells CP that you want to display only the customer-written CP routines loaded using CPXLOAD statements in the system configuration file.

**Userid userid**

tells CP that you want to display only the customer-written CP routines loaded by a specific user ID. The variable *userid* must be a 1-character to 8-character alphanumeric string that represents a virtual machine on your system that is, or was, capable of issuing CPXLOAD commands.

**Usage Notes**

1. To load customer-written CP routines into the system execution space, use the CPXLOAD command (see “CPXLOAD” on page 111). In response to your CPXLOAD command, CP assigns and displays a load identifier (load ID). This load ID is a number between 0 and 2147483647 that identifies the CP routines that you loaded. If you ever want to unload these routines (using the CPXUNLOAD command), you must specify the load ID in the command for CP to know what to unload.

If you do not know the load ID for some CP routines that have already been loaded, you can use one of the following operands:

- ALL (which is the default) to display information about all of the CP routines that have been loaded.
  - EPNAME *name* to display information about only the routines associated with a specific entry point name.
  - USERID SYSTEM to display information about only the routines loaded by the system configuration file at initialization.
  - USERID *userid* to display information about only the routines loaded by a specific user after initialization.
2. To remove customer-written CP routines from the system execution space, use the CPXUNLOAD command (see “CPXUNLOAD” on page 117). After unloading the CP routines, CP frees the load ID and clears the information that was being stored about those CP routines. CP will not reuse this load ID until such time as your system is reinitialized.
  3. For more information about loading customer-written CP routines into the system execution space, see [Loading Dynamically into the System Execution Space in z/VM: CP Exit Customization](#).
  4. The addresses displayed by QUERY CPXLOAD are host logical storage addresses.

If the loaded routine is greater than 4 KB in length, the frames backing the host logical storage are not necessarily contiguous in host real storage.

If you need to know the host real storage address of the routine, enter the host logical storage address on the DISPLAY (Host Storage) command. The DISPLAY command displays the data at that location and also displays the corresponding host real storage address.

**Responses****Note:**

The following sample responses show information for multiple load IDs listed in ascending order. On your screen, multiple load ID responses may be arranged in ascending, descending, or random order. Options shown in the responses are the options specified on the CPXLOAD command and not the options set as a result of the OPTIONS loader directive.

Response 1 (Display by Load ID):

To display information about the customer-written CP routines loaded as identifier 1, enter the following:

```
query cpxload id 1
```

## QUERY CPXLOAD

**1**

Loaded as identifier 1

**2**

File: ZXXKL STUFF Z

**3**

Loaded by: SYSTEM on 10/12/95 at 11:27:59

**4**

CPXLOAD options: MP NOCONTROL DELAY NOLET PERMANENT

**5**

```
Associated entry points and external symbols:
PLOWERTY is at 078DF3A6
ZXXKLM is at 078DF58E
ZXXKLMFF is at 078DF772
ZXXKLMPQ is at 078DF99C
ZXXKLMQU is at 078DF000 module size is 000EF8
```

Ready;

In this example,

**1**

is the load ID assigned to all the routines that were loaded by the file in line **2**.

**2**

is the file identifier of the file:

- Containing the routines that were loaded,
- Which imbedded or included the routines that were loaded, or
- Some combination thereof.
- The *filemode* indicates the file mode of the CP accessed disk at the time of the CPXLOAD.

**3**

tells you the virtual machine responsible for loading the routines and the date and time they were loaded. If the virtual machine is SYSTEM (as in this example), then the routines were loaded by CPXLOAD statements in your system configuration file.

**4**

are the options from the CPXLOAD command or CPXLOAD configuration file statement. It does not include any information from OPTIONS loader directives imbedded in the loaded files. Note that the specified CPXLOAD options are listed as well as the defaults that were taken because an option was not specified.

**5**

lists all of the entry points and external symbols associated with the routine that was loaded. For each entry point or external symbol, CP displays the starting address in the system execution space where you can find the entry point or external symbol, and for modules, CP also displays the module's size.

### Notes:

1. Private control sections (CSECTs) do not have names. If one was loaded, you will see "P/C" listed as the name of the entry point or external symbol. For example:

```
Associated entry points and external symbols:
PVC0CAEA is at 14217208
PVCE3348 is at 142179CC
PVCF3D2E is at 14217D48
P/C is at 14217000 module size is 000E08
```

2. The displayed addresses are host logical storage addresses. See usage note "4" on page 849 for information on how to determine the host real storage address of an external symbol.

Response 2 (Display by Entry Point Name):

To display information about all the customer-written CP routines that have entry points and external symbols that start with a 4-character string of "MINE", enter the following:

```
query cpxload epname mine*
Loaded as identifier 14
File: LCLCMDS TEXT B
Loaded by: SYSTEM on 12/28/95 at 12:12:02
CPXLOAD options: MP CONTROL DELAY LET PERMANENT
Associated entry points and external symbols:
MINEENTR is at 078E0700
MINEDATA is at 078E0174
MINETABL is at 078E0000 module size is 000C00
```

```
Loaded as identifier 32
File: LCLDIAG TEXT B
Loaded by: SYSTEM on 12/28/95 at 23:48:36
CPXLOAD options: MP CONTROL DELAY LET PERMANENT
Associated entry points and external symbols:
MINE0C4 is at 078DF772
MINE0C6 is at 078DF99C
MINESECT is at 078DF000 module size is 000EF8
Ready;
```

Response 3 (Display by User ID):

To display information about all the customer-written CP routines that were loaded after initialization by user MAINT, enter the following:

```
query cpxload userid maint
Loaded as identifier 4
File: HRH CONTROL B
Loaded by: MAINT on 12/18/95 at 21:57:03
CPXLOAD options: MP CONTROL NODELAY NOLET TEMPORARY
Associated entry points and external symbols:
HRD0CAEA is at 0621F208
HR4E3348 is at 0621F9CC
HR8F3D2E is at 0621FD48
HR6438C8 is at 0621F000 module size is 000E08
```

```
Loaded as identifier 7
File: ABEND TEXT R
Loaded by: MAINT on 12/21/95 at 02:13:59
CPXLOAD options: MP CONTROL NODELAY NOLET PERMANENT
Associated entry points and external symbols:
ADENDEP is at 0B760028
ABEND is at 0B760000 module size is 000048
Ready;
```

Response 4 (Display All):

To display information about all the customer-written CP routines that have been loaded onto (and not yet unloaded from) your system, enter the following:

```
query cpxload all
Loaded as identifier 1
File: ZXXKL STUFF Z
Loaded by: SYSTEM on 10/12/95 at 11:27:59
CPXLOAD options: MP NOCONTROL DELAY NOLET PERMANENT
Associated entry points and external symbols:
PLOQWERTY is at 078DF3A6
ZXXKLM is at 078DF58E
ZXXKLMFF is at 078DF772
ZXXKLMFQ is at 078DF99C
ZXXKLMQU is at 078DF000 module size is 000EF8
```

```
Loaded as identifier 4
File: HRH CONTROL B
Loaded by: MAINT on 12/18/95 at 21:57:03
CPXLOAD options: MP CONTROL NODELAY NOLET TEMPORARY
Associated entry points and external symbols:
HRD0CAEA is at 0621F208
HR4E3348 is at 0621F9CC
HR8F3D2E is at 0621FD48
HR6438C8 is at 0621F000 module size is 000E08
```

## QUERY CPXLOAD

```
Loaded as identifier 7
File: ABEND TEXT R
Loaded by: MAINT on 12/21/95 at 02:13:59
CPXLOAD options: MP CONTROL NODELAY NOLET PERMANENT
Associated entry points and external symbols:
ADENDEP is at 0B760028
ABEND is at 0B760000 module size is 000048
```

```
Loaded as identifier 14
File: LCLCMDS TEXT B
Loaded by: SYSTEM on 12/28/95 at 12:12:02
CPXLOAD options: MP CONTROL DELAY LET PERMANENT
Associated entry points and external symbols:
MINEENTR is at 078E0700
MINEDATA is at 078E0174
MINETABL is at 078E0000 module size is 000C00
```

```
Loaded as identifier 32
File: LCLDIAG TEXT B
Loaded by: SYSTEM on 12/28/95 at 23:48:36
CPXLOAD options: MP CONTROL DELAY LET PERMANENT
Associated entry points and external symbols:
MINE0C4 is at 078DF772
MINE0C6 is at 078DF99C
MINESECT is at 078DF000 module size is 000EF8
Ready;
```

Response 5 (Nothing is Loaded):

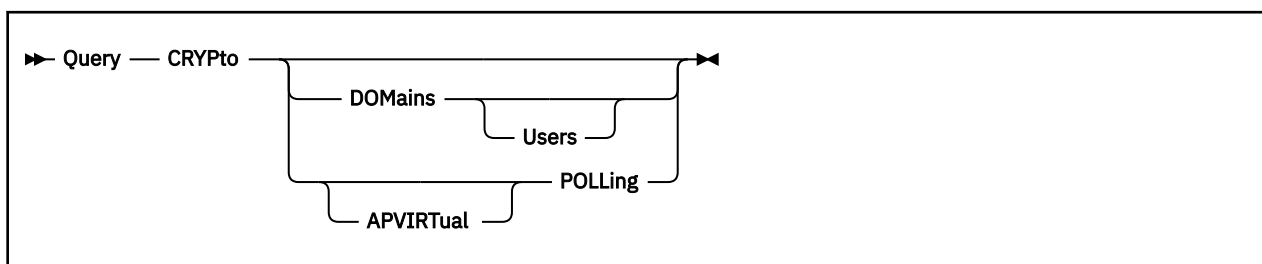
If you issue the QUERY CPXLOAD command and no customer-written CP routines are loaded on your system with CPXLOAD commands or configuration file statements, you will see:

```
query cpxload all
There are no routines in CP storage loaded by CPXLOAD operations
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP013E Conflicting option - *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid {string|number|entry point name} - {*string* | *loadid* | *name*}
- HCP6720I No items found to satisfy selection criteria.

## QUERY CRYPTO



### Authorization

Privilege Class: A, B, C, E

### Purpose

Use QUERY CRYPTO to display the status of the real crypto resources that are available on the system.

### Operands

#### APVIRTual POLLing

requests a display of the polling setting of the shared pool.

#### DOMains

requests a display of installed crypto resources.

#### Users

requests a display of virtual machines that are logged on or disconnected and have access to the system's pool of shared crypto resources.

### Usage Notes

1. The set of crypto resources displayed is limited to the subset available to the logical partition or the virtual machine in which z/VM is running.
2. For information on planning and managing crypto resources on a z/VM system, see [Crypto Planning and Management in z/VM: CP Planning and Administration](#).
3. A VIRTUAL version of this command, QUERY VIRTUAL CRYPTO, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL CRYPTO” on page 1284](#).

### Responses

Response 1:

The following is the response to QUERY CRYPTO if AP crypto instructions are not installed:

```
Crypto Adjunct Processor Instructions are not installed
```

Response 2:

The following is the response to QUERY CRYPTO if AP crypto instructions are installed:

```
Crypto Adjunct Processor Instructions are installed
```

Response 3:

The following is the response to QUERY CRYPTO DOMAINS if crypto adapters are installed:

AP num	AP type	Dom num	Device Status	Config State	Device Assignment
AP 002	CEX7A	Domain 015	operational	online	shared
AP 002	CEX7A	Domain 020	resetting	online	attached to userid
AP 003	UNKN	Domain 015	revoked	online	free
AP 003	UNKN	Domain 020	revoked	online	free
AP 005	CEX8A	Domain 015	operational	offline	free, dedication planned
AP 005	CEX8A	Domain 020	operational	offline	free

**Note:** The column headings are not included in the output and are shown here for clarity.

The output contains the following fields:

**apnum**

The *apnum* field indicates the three-digit crypto adapter number in decimal.

**aptype**

The *aptype* field indicates the crypto adapter type and mode. For dedicated resources, the value indicates the type and mode of the physical resource. For shared resources, the value indicates the maximal common subset of crypto express adapter capabilities that is available in the shared pools of all systems in the user's relocation domain that have the same mode. Shared crypto resources must be configured in accelerator or CCA-coprocessor mode. The value of the *aptype* field can be one of the following:

**CEX3C**

Crypto Express3 configured in CCA coprocessor mode

**CEX3A**

Crypto Express3 configured in accelerator mode

**CEX4C**

Crypto Express4 configured in CCA coprocessor mode

**CEX4A**

Crypto Express4 configured in accelerator mode

**CEX4P**

Crypto Express4 configured in EP11 coprocessor mode

**CEX5C**

Crypto Express5 configured in CCA coprocessor mode

**CEX5A**

Crypto Express5 configured in accelerator mode

**CEX5P**

Crypto Express5 configured in EP11 coprocessor mode

**CEX6C**

Crypto Express6 configured in CCA coprocessor mode

**CEX6A**

Crypto Express6 configured in accelerator mode

**CEX6P**

Crypto Express6 configured in EP11 coprocessor mode

**CEX7C**

Crypto Express7 configured in CCA coprocessor mode

**CEX7A**

Crypto Express7 configured in accelerator mode

**CEX7P**

Crypto Express7 configured in EP11 coprocessor mode

**CEX8C**

Crypto Express8 configured in CCA coprocessor mode

**CEX8A**

Crypto Express8 configured in accelerator mode

**CEX8P**

Crypto Express8 configured in EP11 coprocessor mode

**NONE**

There are no crypto resources attached to the shared pool.

**UNKN**

z/VM is unable to determine the type of crypto adapter.

***domnum***

is the three-digit domain number in decimal. If the resource is shared, then 001 is assigned to the virtual domain number. If the resource is dedicated, then the actual hardware domain number of the resource is assigned.

***device\_status***

can be any of the following:

**operational**

indicates that the crypto resource is installed and operational.

**checkstop**

indicates that the crypto resource is in a checkstop condition and is unavailable.

**deconfigured**

indicates that the adapter is deconfigured and unavailable. This could result from VARY OFF CRYPTO when the environment in which CP is running supports AP reconfiguration, or from an operation performed on the hardware maintenance console (HMC).

**busy**

indicates that the adapter is temporarily busy initializing or doing error recovery.

**resetting**

indicates that the resource is being reset.

**revoked**

indicates the resource was detected by CP, but has since been unassigned from the configuration. If the resource is added back into the configuration, the updated status will be reported. Otherwise, when CP no longer detects this resource in the configuration, it will not be reported in Q CRYPTO output.

**unsupported**

indicates the crypto resource status is unsupported by CP.

***config\_state***

CP's logical view of the resource state, as controlled by VARY CRYPTO. Can be any of the following:

**online**

indicates that the crypto resource is online and available for use.

**offline**

indicates that the crypto resource is offline.

***device\_assignment***

can be any of the following:

**free, dedication planned**

indicates that the crypto resource is not in use, however, it has been specified on a CRYPTO APDED statement in the online user directory.

**attached to *userid***

indicates that the crypto resource is dedicated to a logged on virtual machine.

**free**

indicates that the crypto resource is not in use.

**shared**

indicates that the crypto resource is attached to the system for shared use.

This response is repeated for each crypto resource installed on the system.

Response 4:

## QUERY CRYPTO

The following is the response to QUERY CRYPTO DOMAINS if no crypto adapters are assigned to the zVM system.

```
No AP Crypto Domains are available
```

Response 5:

The following is the response to QUERY CRYPTO DOMAINS USERS.

- The existing output for QUERY CRYPTO DOMAINS is displayed, followed by a blank line.
- If there are no crypto resources available for sharing, the following message is displayed:

```
There are no AP crypto resources available for shared use.
```

- If there are virtual machines that are logged on or disconnected and have access to the shared crypto resources on the system, the following header is displayed, followed by a list of user IDs with up to six user IDs per line:

```
Shared-Crypto Users:
```

- If there are no virtual machines logged on or disconnected that have access to the shared crypto resources on the system, the following message is displayed:

```
There are no shared-crypto users.
```

Response 6:

The following is the response to QUERY CRYPTO POLLING.

If CRYPTO APVIRTUAL POLLING is set ON, the following message is displayed:

```
Shared-crypto polling is ON
```

If CRYPTO APVIRTUAL POLLING is set OFF, the following message is displayed:

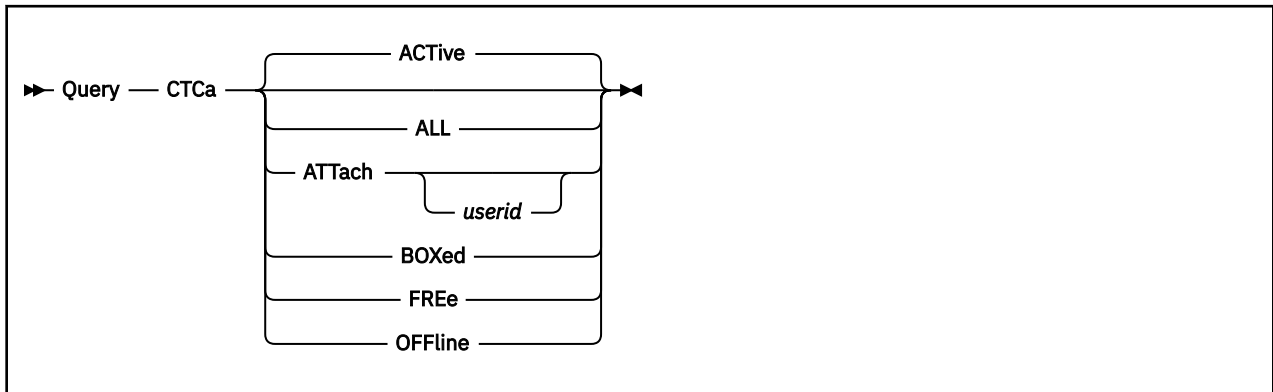
```
Shared-crypto polling is OFF
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra options starting with *option*
- HCP1014E A required option is missing.



## QUERY CTCA



### Authorization

Privilege Class: B

### Purpose

Use QUERY CTCA to display the status of real channel-to-channel adapters.

### Operands

#### ACTive

indicates active channel-to-channel adapters. An active device is one that is in use by a user or the system and is specified as neither FREE nor OFFLINE.

#### ALL

indicates all channel-to-channel adapters attached to the system. The status is displayed in the order of ACTIVE, FREE, and OFFLINE. You receive the same response as you would receive if you entered all of the following commands:

```

query ctca active
query ctca free
query ctca offline
  
```

#### ATTach *userid*

indicates the channel-to-channel adapters that are dedicated to a user on the system. If you specify *userid*, only adapters attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all adapters dedicated to any user on the system. Attached devices are always active.

#### BOXed

indicates the channel-to-channel adapters that are unavailable (usually because of a hardware condition). A boxed adapter can also be active, free, or offline.

#### FREe

indicates all free channel-to-channel adapters. A free adapter is one that is not currently in use by a user or the system and is specified as neither ACTIVE nor OFFLINE.

#### OFFline

indicates all offline channel-to-channel adapters. An offline adapter is one that is specified as neither ACTIVE nor FREE.

## Usage Notes

- A VIRTUAL version of this command, QUERY VIRTUAL CTCA, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL CTCA” on page 1287](#).

## Responses

If the system is using a channel-to-channel adapter that is in an intervention-required state and you enter QUERY CTCA, you see a response like this:

```
CTCA 0300 has intervention required.
```

0300 is the real device number of the channel-to-channel adapter that needs the operator's intervention.

**Note:** This response will be preceded by one of the following.

Response 1:

If you enter QUERY CTCA, QUERY CTCA ACTIVE, or QUERY CTCA ATTACHED and there is a channel-to-channel adapter attached to the system, you see something like this:

```
q ctca active
CTCA 0480 ATTACHED TO PVM      0480
CTCA 0481 ATTACHED TO PVM      0481
CTCA 0482 ATTACHED TO RSCS     0482
CTCA 0483 ATTACHED TO RSCS     0483
CTCA 0484 ATTACHED TO PVM      0485
CTCA 0485 ATTACHED TO PVM      0484
CTCA 0486 ATTACHED TO VTAM     0486
CTCA 0700 ATTACHED TO TCPIP    0500
CTCA 0701 ATTACHED TO TCPIP    0501
Ready;
```

The first number is the real device number of the channel-to-channel adapter. The ATTACHED operand might also read BOX/ATTC, which means that although the device is attached and dedicated to a user, it is unavailable. PVM, RSCS, TCPIP, and VTAM are user IDs to which the CTCAs are dedicated. The last number is the virtual device number which each virtual machine uses to refer to a CTCA.

Response 2:

If you enter a QUERY CTCA FREE command and there is a free channel-to-channel adapter attached to the system, you see something like this:

```
q ctca free
CTCA 0487 FREE      , CTCA 04A1 FREE      , CTCA 04A2 FREE      , CTCA 04A3 FREE
CTCA 04A4 FREE      , CTCA 04A5 FREE      , CTCA 04A9 FREE      , CTCA 04AA FREE
CTCA 04AB FREE      , CTCA 04BC FREE      , CTCA 04BD FREE      , CTCA 04BE FREE
CTCA 04BF FREE
Ready;
```

This response shows the real device number for each free CTCA.

The response for a QUERY CTCA OFFLINE or QUERY CTCA BOXED command is in the same format as the response for a QUERY CTCA FREE command.

Response 3:

When you enter a QUERY CTCA ALL command, CP displays the same responses as it would display if you had entered the QUERY CTCA ACTIVE, QUERY CTCA FREE, and QUERY CTCA OFFLINE commands.

Response 4:

If CP cannot find a channel-to-channel adapter of the type you specify, you see a message like this:

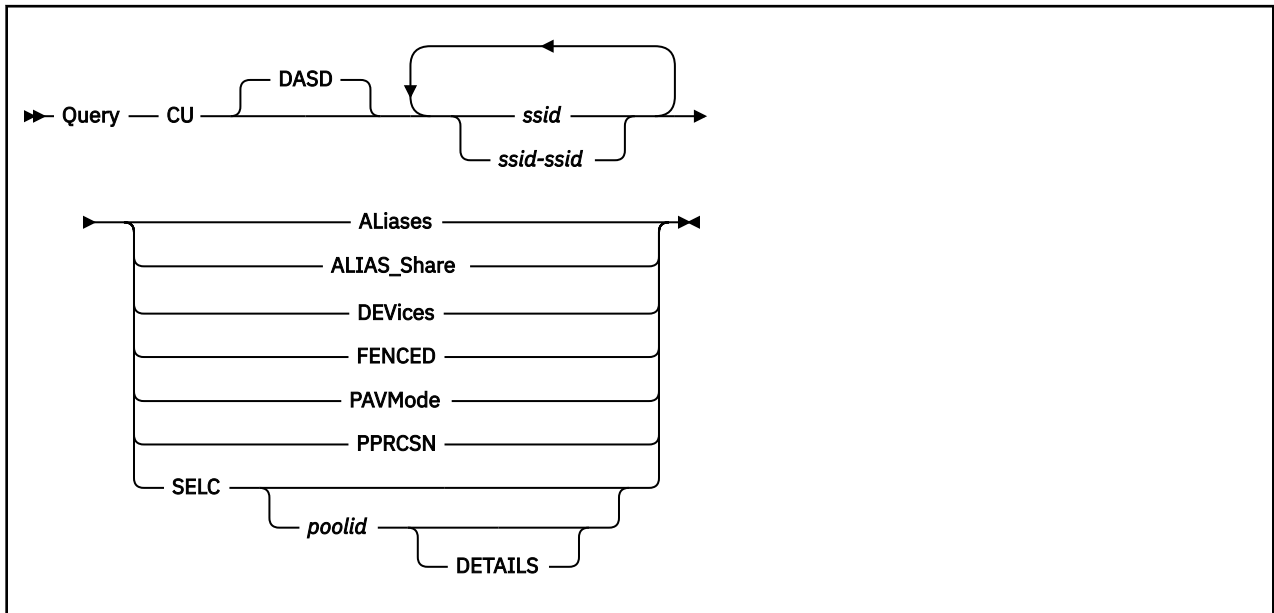
```
q ctca boxed
A BOXED CTCA was not found.
Ready;
```

In this response, BOXED may be replaced with `active`, `attached`, `offline`, or `free`, depending on what you specified on the command.

### Messages

- HCP003E Invalid option - *option*
- HCP006E Invalid device type - *rdev*
- HCP007E Invalid userid - *userid*
- HCP263E Too many operands were supplied for this command.

## QUERY CU



### Authorization

Privilege Class: B

### Purpose

Use QUERY CU to display selected information about a DASD control unit.

#### Notes:

- The QUERY CU command applies only to first-level z/VM systems. When z/VM is running as a guest of another z/VM system, the guest z/VM system does not control the operation of the control unit and cannot query the information about the control unit. If the command is issued to a guest z/VM system, the command is rejected.
- The control unit that is associated with an EDEVICE HyperPAV pool has a subsystem identifier (SSID) that cannot be referenced by the QUERY CU command and cannot be interrogated by the QUERY CU command.

### Operands

#### *ssid*

#### *ssid-ssid*

is the subsystem identifier (SSID) of a control unit as established in a control unit during its installation. The QUERY CU command operates on the control unit that is identified. The variable *ssid* must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single subsystem identifier, a list, a range, or any combination thereof.

#### **Aliases**

requests the display of PAV or HyperPAV alias devices in a control unit.

#### **ALIAS\_Share**

requests the display of minidisk I/O's and paging I/O's relative shares of and entitlements to system-attached HyperPAV aliases.

#### **DEVICES**

requests the display of all devices in a control unit.

**FENCED**

requests the display of all devices in soft fence state in a control unit.

**PAVMode**

requests the display of information about the Parallel Access Volume (PAV) mode of a control unit. This information includes the current PAV mode of the control unit and the allowed PAV mode of the control unit. (See also the SET CU command.)

**PPRC SN**

requests to display the status of PPRC Summary Notifications for the given control units.

**SELC**

specifies that the space-efficient logical configuration is desired.

**poolid**

specifies the requested space-efficient pool number. If omitted, all pools are displayed.

**DETAILS**

displays additional information regarding persistent FlashCopy relationships that were established on minidisks owned by the current user during the current system IPL session.

**Usage Notes**

1. The z/VM system programmer and the DASD management programmer must work together to make sure each Logical Subsystem (LSS) that is being used on the system has a unique SSID.

**Responses**

Response 1:

To query the devices in control units 4343, 4344, and 4345, enter the following:

```
q cu 4343-4345 dev
DASD CU 4343 DEVICES:
4300 4301 4302 4303 4304 4305 4306 4307
4308 4309 430A 430B 430C 430D 430E 430F
4310 4311 4312 4313 4320 4321 4322 4323
4324
DASD CU 4344 DEVICES:
4410 4411 4412 4413 4420 4421 4422 4423
DASD CU 4345 DEVICES:
4501 4502 4503 4504
Ready;
```

Response 2:

To query the alias devices (PAV or HyperPAV) in control units 4343, 4344, and 4345, enter the following:

```
q cu 4343-4345 a1
DASD CU 4343 ALIASES:
4304 4305 4306 4307 4308 4309 430A 430B
430C 430D 430E 430F 4311 4312 4313 4321
4322 4323 4324
DASD CU 4344 ALIASES:
4411 4412 4413 4421 4422 4423
DASD CU 4345 ALIASES:
4502 4503 4504
Ready;
```

Response 3:

To query the PAV mode of control units 4343, 4344, 4345, and 4346, enter the following:

```
q cu 4343-4346 pavmode
DASD CU 4343 CURRENTLY HYPERPAV, HYPERPAV ALLOWED
DASD CU 4344 CURRENTLY PAV, PAV ALLOWED
DASD CU 4345 CURRENTLY NOPAV, PAV ALLOWED
DASD CU 4346 DOES NOT EXIST, HYPERPAV ALLOWED
Ready;
```

## QUERY CU

This response indicates that CU 4343 is operating at its fully allowed HYPERPAV capability; CU 4344 is operating at its fully allowed PAV capability; CU 4345 is operating without alias devices but is allowed to operate in PAV mode if and when the requisite feature is installed for PAV operation; and CU 4346 is not in the configuration, but a CU statement in the system configuration file or an earlier SET CU command allowed HYPERPAV mode. For CU 4346, the allowance of HYPERPAV mode will take effect if and when the control unit is added to the configuration (for example, with a DEFINE CU command).

Response 4:

If you query a control unit to display its devices but there are none, the following response is displayed:

```
q cu 4347 dev
DASD CU 4347 HAS NO ASSOCIATED DEVICES
Ready;
```

Response 5:

If you query a control unit to display its alias devices but there are none, the following response is displayed:

```
q cu 4347 al
DASD CU 4347 HAS NO ASSOCIATED ALIASES
Ready;
```

Response 6:

If you query a control unit to display its PAV mode but that control unit type does not support alias devices, the following response is displayed:

```
q cu 4348 pavmode
DASD CU 4348 PAV FUNCTIONS DO NOT APPLY
Ready;
```

Response 7:

If you query a control unit and it does not exist in your configuration, the following response is displayed:

```
q cu 4349 dev
DASD CU 4349 DOES NOT EXIST
Ready;
```

**Note:** If you specify a range, and a control unit in that range does not exist, that control unit is not shown in the response unless it is specified in a CU statement in the system configuration file or in an earlier SET CU command.

Response 8:

If you query a control unit for a space-efficient logical configuration, the following response is displayed:

```
q cu dasd 0101 selc
SSID 0101: TYPE=002107 MODEL=921 SN=0000004191
SSID 0101: POOLID RPOSWARN POOLWARN STATUS...
SSID 0101:      0      0%      0% CKD RPOSCONF REPWARN REPFULL
SSID 0101:      1      0%      0% CKD RPOSCONF REPWARN REPFULL
SSID 0101:      3      0%      0% FBA RPOSCONF REPWARN REPFULL
SSID 0101:      4      0%      0% FBA RPOSCONF REPWARN REPFULL
Ready; T=0.01/0.01 11:02:10
```

where the STATUS... data will be:

Status	Meaning
CKD	- This is a CKD volume
FBA	- This is an FBA volume
RPOSCONF	- The extent pool repository is configured
REPWARN	- The repository is at the warning percentage

Status	Meaning
REPFULL	- The repository is full
POOLWARN	- The extent pool is at the warning percentage
POOLFULL	- The extent pool is full

If the command is issued with the *poolid* DETAILS operands, the following additional information is displayed:

```
query cu dasd 0101 selc 1 details
SSID 0101: TYPE=002107 MODEL=921 SN=0000004191
SSID 0101: POOLID RPOSWARN POOLWARN STATUS...
SSID 0101:      1      0%      0% CKD RPOSCONF REPWARN REPFULL
SSID 0101: POOL=1639449/1943298 84%
SSID 0101: RPOS=0/0
SSID 0101: GUAR=0/0
Ready; T=0.01/0.01 11:01:21
```

For POOL=xxx/yyy zzz%, RPOS=xxx/yyy zzz%, and GUAR=xxx/yyy zzz%, the xxx value is the currently allocated extent space, yyy is the max size of the extent, and zzz is just x/y. These are for the extent pool, extent pool repository, and guaranteed space values.

Response 9:

When an SSID has enabled PPRC Summary Notifications, the following response is displayed:

```
DASD CU ssid IS ENABLED FOR PPRCSN
```

Response 10:

When an SSID has disabled PPRC Summary Notifications, the following response is displayed:

```
DASD CU ssid IS DISABLED FOR PPRCSN
```

Response 11:

If you query a control unit to display its devices in soft fence state, the following response is displayed:

```
q cu 4343 fenced
DASD CU 4343 DEVICES IN SOFT FENCED STATE:
4300 4301 4302 4303 4304 4305 4306 4307
4308 4309 430A 430B 430C 430D 430E 430F
4310 4311 4312 4313 4320 4321 4222 4323
4324
Ready;
```

Response 12:

If you query a control unit to display its devices in soft fence state but there are none, the following response is displayed:

```
q cu 4344 fenced
DASD CU 4344 HAS NO DEVICES IN SOFT FENCED STATE.
Ready;
```

Response 13:

To query the alias-shares and entitlements in control unit 4343, when PAGING\_share is 200, MDISK\_share is 100, and the total number of system-attached HyperPAV aliases is 17:

```
q cu 4343 alias_share
DASD CU 4343
MDISK ALIAS: SHARE    100 ENTITLEMENT    6
PAGING ALIAS: SHARE   200 ENTITLEMENT   11
Ready;
```

The minidisk entitlement calculation is:

```
(100 / (100 + 200)) * 17 = 17 / 3 = 5.67 (rounded up to 6)
```

The paging entitlement calculation is:

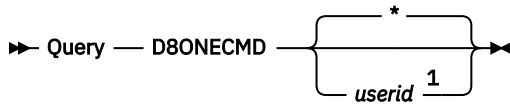
```
(200 / (100 + 200)) * 17 = 34 / 3 = 11.33 (rounded down to 11)
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP319E Error processing subsystem command - *cmd* for DASD *vdev*
- HCP332E Invalid control unit type - *vdev*
- HCP6004E This command is not supported on your system.
- HCP6869E DASD CU *ssid* devices soft fence state could not be determined because I/O error occurred.
- HCP6869E DASD CU *ssid* does not support soft fence.



## QUERY D8ONECMD



Notes:

<sup>1</sup> The `userid` variable is not permitted for privilege class G.

### Authorization

Privilege Class: A, C, E, G

### Purpose

Use QUERY D8ONECMD to determine the D8ONECMD settings for your own virtual machine or another user's virtual machine.

### Operands

\*

indicates that you wish to display your own virtual machine's D8ONECMD settings. \* is the default.

*userid*

is the user ID of the virtual machine for which you want to display the D8ONECMD settings.

### Usage Notes

1. The initial D8ONECMD settings for your virtual machine are determined by options set in your user directory entry.
2. You can use the class G SET D8ONECMD command to change your D8ONECMD settings. You can use the class A SET D8ONECMD command to change another virtual machine's D8ONECMD settings. For more information, see [“SET D8ONECMD” on page 1482](#).

### Responses

If you enter QUERY D8ONECMD, you receive one of the following responses:

Response 1:

```
userid can enter multiple commands with a DIAGNOSE X'08'.
This setting is locked against changes.
```

*userid*

represents the user ID whose D8ONECMD settings are being displayed.

This response indicates that you can process a DIAGNOSE code X'08' containing multiple CP commands with no restrictions. The second line of the response indicates that only a privileged user who is also not the command issuer can change the settings.

Response 2:

```
CP will log userid's attempts to issue
multiple commands with a DIAGNOSE X'08'.
This setting is locked against changes.
```

### ***userid***

represents the user ID whose D8ONECMD settings are being displayed.

This response indicates that any attempt to process a DIAGNOSE code X'08' containing multiple CP commands will not fail but will cause a message to be sent to operations personnel. The last line of the response indicates that only a privileged user who is also not the command issuer can change the settings.

Response 3:

```
CP will block userid's attempts to issue
multiple commands with a DIAGNOSE X'08'.
This setting is locked against changes.
```

### ***userid***

represents the user ID whose D8ONECMD settings are being displayed.

This response indicates that any attempt to process a DIAGNOSE code X'08' containing multiple CP commands will fail and will cause a message to be sent to operations personnel. The last line of the response indicates that only a privileged user who is also not the command issuer can change the settings.

Any of the above messages may also have the following line as the second line of the response:

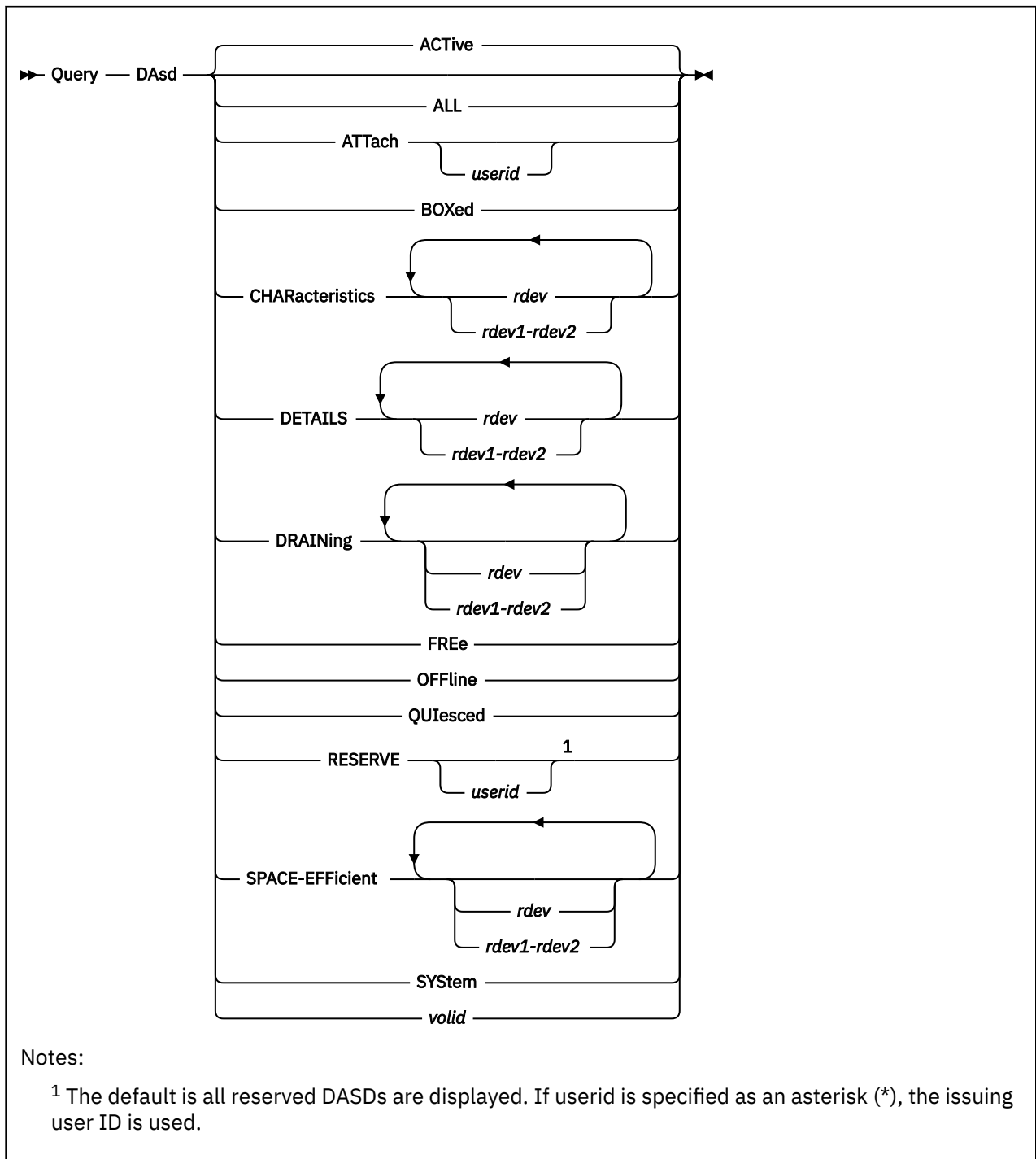
```
This setting is not locked against changes.
```

This response indicates that a privileged user can change the command filtering level for the selected virtual machine (even if the target virtual machine is the command issuer's virtual machine).

## **Messages**

- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*
- HCP045E *userid* not logged on

# QUERY DASD



## Authorization

Privilege Class: B

## Purpose

Use QUERY DASD to display the status of real direct access storage devices.

## Operands

### ACTive

indicates all active direct access storage devices. An active device is one that is in use by a user or the system and is specified as neither FREE nor OFFLINE. This is the default.

### ALL

indicates all direct access storage devices attached to the system. The status is displayed in the order of ACTIVE, FREE, and OFFLINE. You receive the same response as you would receive if you entered all of the following commands:

```
query dasd active
query dasd free
query dasd offline
```

### ATTach *userid*

indicates the DASDs that are dedicated to a user on the system. If you specify *userid*, only devices attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all devices dedicated to any user on the system. Attached devices are always active.

### BOXed

indicates the DASDs that are unavailable (usually because of a hardware condition). A boxed device may also be active, free, or offline.

### CHARacteristics

displays a hexadecimal block that contains device configuration and characteristics information. To get formatted output, use the IOEXPLOR utility.

For further details on the information contained in the output, consult your storage hardware vendor.

### DETAILS

displays detailed information about real DASD.

#### *rdev*

#### *rdev1-rdev2*

is the real device number (*rdev*), a list of real device numbers (*rdev rdev rdev*), or a range of real device numbers (*rdev1-rdev2*). A real device can be entered as a 4-digit hexadecimal number between X'0000' and X'FFFF'. For the CHARacteristics and DETAILS operands, a real device in the active configuration can be entered as a 4-digit hexadecimal number between X'0000' and X'FFFF' and all real devices can be specified as a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured. A range of devices cannot span subchannel sets.

For DASD attached to a noncaching storage control, the information provided is:

- Control unit type and model
- Device type and model
- Volume serial number (VOLSER).

For DASD attached to a caching storage control (see Usage Notes), the following additional information is provided:

- Caching status for subsystem and device
- Nonvolatile storage status
- DASD fast write device status
- Cache fast write subsystem status
- Pinned data status
- Channel connection address (CCA)
- Director to device connection (DDC) address
- Duplex information
- Concurrent Copy (CONCOPY) session information

- Peer-to-Peer Remote Copy (PPRC) information
- Soft Fence state
- Host Access information.

**DRAINing**

specifies the devices for which status is to be displayed, if they are currently draining. If no real device list is supplied, all DASD is examined.

**FREe**

indicates all free DASDs. A free device is one that is not currently in use by a user or the system and is specified as neither ACTIVE nor OFFLINE.

**OFFline**

indicates all offline DASDs. An offline device is one that is specified as neither ACTIVE nor FREE. The offline response will also show offline devices which are quiesced.

**QUIesced**

indicates all quiesced DASD. A quiesced device is an offline device which cannot be varied online until a resume request is received for the device.

**RESERVE *userid***

indicates all reserved DASD defined as full-pack minidisks. If you specify *userid*, only devices attached to that user's virtual machine are indicated.

**SPACE-EFFicient**

specifies that information regarding space-efficient real volumes is to be listed. The displayed results identify the real device number, volser, space-efficient pool number, pool size and usage statistics.

**SYStem**

indicates the DASDs in use by the system. A system device is always active.

***valid***

specifies the volume identification of a DASD whose status is to be displayed. If a volume has a nonstandard label, the volume ID reported is unpredictable.

**Usage Notes**

1. The DASD DETAILS function is not supported for caching models of the 3880 storage control (Models E21, G21, J21, H21, D23, E23, G23, J23, and H23) and produces error message HCP332E instead of a response. For these devices, you should use the QUERY CACHE *rdev* command.
2. For the DASD RESERVE function, only DASDs defined as full-pack minidisks with the SHARED option and active or pending reserves are displayed.
3. The DASD RESERVE function can be used to determine the reserve status of system DASD.
4. For the DASD RESERVE function, if either the RESERVED BY SYSTEM response or the RESERVE PENDING FOR SYSTEM response appears consistently for the same device over a period of time, the response could be indicating a problem.
5. Users who have been defined in QUICKDSP cannot afford an I/O delay, so DASD DETAILS will not issue Soft Fence and Query Host Access related I/O.
6. The DASD DETAILS function will not issue Soft Fence and Query Host Access related I/O to a device that is already quiesced due to an active HYPERSWAP QUIESCE function. This is because the I/O queued and the issuing user will be hung when the target device is in HYPERSWAP processing.
7. Host Access information can be requested only for dedicated DASD and fullpack minidisks. If the DASD DETAILS function is issued from a second-level system to get Host Access information for the specified device, the information can be obtained only when the underlying first-level disk is a dedicated DASD or a fullpack minidisk. Additionally, Host Access information can be obtained from a base device or a PAV alias device, but not a HyperPAV alias device.
8. A VIRTUAL version of this command, QUERY VIRTUAL DASD, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL DASD” on page 1288](#).

## Responses

### Response 1:

If you enter the QUERY DASD *valid* command when there are no online DASDs, you receive one of the following responses:

```
DASD valid was not found.
```

- or -

```
DASD valid offline, label queried may not be currently on the DASD
```

**Note:** The second response is also issued for the secondary volume of a duplex pair, because secondary volumes can be offline. In this case the label queried is in fact on the DASD, because the secondary volume is a copy of the primary (unless the pair is suspended). Another response describing the primary volume appears with this response.

### Response 2:

If the system is using any DASD that is in an intervention-required state and you enter QUERY DASD, you receive a response that looks like this:

```
DASD 181B has intervention required.
```

181B is the real device number.

**Note:** This response will be preceded by one of the following.

### Response 3:

The following response is displayed if the real device is attached to the system for access as user minidisks or for system paging, spooling, or temporary disk space.

```
q dasd
DASD 0280 CP SYSTEM XAUSR1 49
DASD 0281 CP SYSTEM XAUSR2 24
DASD 0282 CP SYSTEM XAUSR3 49
DASD 0283 CP SYSTEM XAUSR4 34
DASD 0284 CP SYSTEM XAUSR5 42
DASD 0285 CP SYSTEM XAUSR6 20
DASD 0A01 CP OWNED ESAP01 3
DASD 0A02 CP OWNED ESAP02 1
DASD 0A03 CP OWNED ESAP03 1
DASD 0A04 CP OWNED ESAP04 1
DASD 0A05 CP OWNED ESAP05 1
DASD 0C08 CP SYSTEM ESAP7 1
DASD 0C09 CP SYSTEM ESAP6 1
DASD 1800 CP SYSTEM USPK1 4
DASD 1801 CP SYSTEM USPK2 2
Ready;
```

The first number listed is the real device number of the DASD. SYSTEM indicates the device is allocated to the system for use as users' minidisks. OWNED indicates the device is used by the system for paging and spooling activity. XAUSR1, ESAP01, and the rest are volume labels of the devices. The last number indicates the number of links to minidisks contained on the volume.

### Response 4:

The following response is also displayed if the real device is attached to the system for access as user minidisks or for system paging, spooling, or temporary disk space. This response is longer than the previous response because the device may be reserved (shared option is on), it may have a reserve pending, or it is currently reserved.

```
q dasd active
DASD 181A CP SYSTEM ESAP2C SHARED
DASD 181B CP OWNED ESAP2F RESERVE PENDING FOR USER DAVID
DASD 181C CP OWNED ESAP30 RESERVED BY SYSTEM
Ready;
```

Response 5:

If the device is dedicated to a user, you see a message like this:

```
q dasd
DASD 0820 ATTACHED TO OLIVER 01FC R/O XAUSR4
Ready;
```

This means the DASD at the real device number 0820 is attached to OLIVER's virtual machine as the virtual device 01FC. The DASD's volume identifier is XAUSR4. R/O means the user OLIVER has read-only access to the DASD. R/O might also read R/W, which means the user had read-write access. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

Response 6:

If you enter QUERY DASD DRAINING, you see something like this:

```
q dasd draining
DASD Vol-ID Rdev Draining: Page LInks SPool TDisk
0C02 DR4209 0C02 Yes Yes Yes Yes
0C04 DR5703 0C04 No No Yes No
Ready;
```

This response shows the device number and volume identifier of the DASD, as well as which operations are prevented on it. The second line indicates that CP will not write pages to the DASD at 0C02, allow LINK commands to minidisks on it, allocate space on it for new spool records, or allocate temporary disk requests on it. The third line indicates that CP will not allocate space for new spool records on the DASD at 0C04.

Response 7:

If you enter QUERY DASD DETAILS and the device is attached to a noncaching storage control, you see the following response:

```
rdev CUTYPE = cccc-mm, DEVTYPE = dddd-mm, VOLSER = vvvvvv,
CYLS = nnnnnnnn
```

If you enter QUERY DASD DETAILS for devices attached to a caching storage control (see Usage Notes), you see something like this:

```
query dasd details 1801
1801 CUTYPE = 3990-EC, DEVTYPE = 3390-06, VOLSER = USPK2, CYLS = 3339
CACHE DETAILS: CACHE NVS CFW DFW PINNED CONCOPY
-SUBSYSTEM Y Y Y - N N
-DEVICE Y - - N N N
DEVICE DETAILS: CCA = 01, DDC = 01, SS = 00, ACTIVE = Y, ESE = Y
DUPLICATE DETAILS: Secondary, OTHER DEVICE = 1800, Active duplex
PAV DETAILS: Base volume with 02 alias volumes
PPRC DETAILS: xxxxxx VOLUME
CU DETAILS: SSID = nnnn, CUNUM = nnnn
SERIAL NUMBER: 01XX-12345
Ready;
```

The first line of the response tells you the DASD at real device number 1801 has a control unit type of 3990-EC, a device type of 3390-06, and a volume serial number of USPK2 and contains 3339 cylinders.

In the control unit type and the device type, the last two digits are a hex code for the model and mode. See the storage control documentation to determine the model and mode for the hex code given.

The second line identifies the columns in the CACHE DETAILS portion of the response:

Details	Meaning
CACHE	tells you about caching
NVS	tells you about nonvolatile storage
CFW	tells you about the cache fast write subsystem

Details	Meaning
DFW	tells you about the DASD fast write device
PINNED	tells you about pinned data
CONCOPY	tells you about Concurrent Copy sessions.

In the third and fourth lines of the response, Y and N indicate the following status:

Column	Y	N
CACHE (See note)	The subsystem cache is available, or the cache is activated for the device.	The subsystem cache is not available, or the cache is not activated for the device.
NVS (See note)	The nonvolatile storage for the subsystem is available.	The nonvolatile storage for the subsystem is not available.
CFW	The cache fast write function is active for the subsystem.	The cache fast write function is not active for the subsystem.
DFW	The DASD fast write function is active for the device.	The DASD fast write function is not active for the device.
PINNED	The subsystem has pinned data for one or more devices, or a data track has become pinned.	There is no pinned data for any of the devices in the subsystem, or there are no pinned data tracks.
CONCOPY	One or more Concurrent Copy (CONCOPY) sessions are associated with the subsystem., or tracks on the device are associated with a CONCOPY session.	No CONCOPY session is associated with the subsystem, or no tracks on the device are associated with a CONCOPY session.

**Note:** For subsystem cache, the status can be one of the following:

- Y subsystem caching active
- N subsystem caching inactive
- T subsystem caching terminated
- M subsystem caching disabled for maintenance
- F subsystem caching pending inactive, destage has failed

For subsystem NVS, the status can be one of the following:

- Y subsystem NVS active
- N subsystem NVS inactive
- T subsystem NVS terminated
- M subsystem NVS disabled for maintenance

The fifth line of the response, DEVICE DETAILS, tells you the channel connection address (CCA = 01), the address of the director-to-device connection (DDC = 01), the device subchannel set number (SS = 00), if the device is in the active subchannel set (ACTIVE = Y), and if the device is an extent space efficient volume (ESE = Y). The channel connection address is the channel connection address at the channel control interface.

The director-to-device connection (DDC) is the director-to-device connection at the control unit interface. The DDC can be used to identify the physical location of a device. Because the 3990 Models 3 and 6 can swap devices around, the CCA no longer has a one-to-one correspondence with the DDC. The DDC can be used to locate a physical device that had previously been swapped because of a suspended primary



device in a duplex pair. When the real device is offline, or is connected via FICON, there is no DDC data, and hyphens (--) are displayed.

The sixth line of the response, DUPLEX DETAILS, tells you about the duplex control operations for the DASD at the specified device number. 1800 is the real device number of the volume that is the other volume of the duplex pair. If the other volume could not be located, hyphens (--) are displayed. The full response above tells you the following:

- The DASD at the real device number 1801 is the secondary device in the duplex pair.
- The other device in the duplex pair is at 1800.
- The duplex operation is active.

If the specified DASD was the primary device in the duplex pair, Secondary would read Primary.

In the above example, the status of the Dual Copy pair is `Active duplex`. Other status conditions that could be indicated here instead, would be `Pending duplex` or `Suspended duplex`.

The seventh line of the response, PAV DETAILS or HYPERPAV DETAILS, tells you whether the queried DASD is a Parallel Access Volume or a HyperParallel Access Volume. If the device is a base Parallel Access Volume, as in the above example, the response indicates the number of alias volumes currently associated with that base. If the device is an alias Parallel Access Volume, the response indicates the real device number of the associated base:

```
PAV DETAILS: Alias volume associated with base rdev
```

If the device is a base HyperParallel Access Volume, the response indicates the associated pool that the device is in:

```
HYPERPAV DETAILS: Base volume in Pool 2
```

If the device is an alias HyperParallel Access Volume, the response indicates the associated pool that the device is in:

```
HYPERPAV DETAILS: Alias volume in Pool 2
```

If the queried DASD is not a Parallel Access Volume or a HyperParallel Access Volume, then the PAV DETAILS and HYPERPAV DETAILS lines are not included in the response.

The eighth line of the response, PPRC DETAILS, tells you the queried active Peer-to-Peer Remote Copy mode. In the above sample, `xxxxxxx` is one of the modes:

#### **PRIMARY**

The volume from which PPRC makes a copy.

#### **SECONDARY**

The volume that maintains a consistent copy of the primary volume.

#### **CASCADING INTERMEDIATE**

The volume that can be both a PPRC primary and secondary volume, allowing a multi-hop operation. In a multi-hop operation, PPRC uses a primary volume to send to a secondary volume and that secondary volume operates as a PPRC Extended Distance primary volume, sending to a tertiary volume many miles away. Such a configuration supports disaster recovery operations.

#### **MT PRIMARY**

The volume is a Multi-Target PPRC primary.

If the queried DASD is not an active PPRC volume, the PPRC DETAILS line is not included in the response.

The ninth line of the response, CU DETAILS, tells you additional information about the control unit associated with the DASD. In the above sample, `SSID = nnnn` is the subsystem identifier associated with the control unit and `CUNUM = nnnn` is the control unit number associated with the control unit.

The tenth line of the response, SERIAL NUMBER, tells you the serial number of the associated DASD. This line is constructed as follows:

- The manufacturer indicator - two bytes ('01' for IBM, 'XX' for non-IBM)

## QUERY DASD

- The plant of manufacture - two bytes
- A hyphen (-)
- The rightmost five digits of the storage facility sequence number.

Response 8:

If you enter QUERY DASD DETAILS for an FBA DASD, such as a 9336 model 20, you see something like this:

```
query dasd details 600
0600 CUTYPE = 6310-80, DEVTYPE = 9336-10, VOLSER = L3RES, CYLS = 16
      BLKS = 34504
Ready;
```

This response tells you the DASD at real device number 600 has a control unit type of 6310-80, a device type of 9336-10, a volume serial number of L3RES, and contains 34,504 512-byte blocks. The number of cylinders displayed (e.g., CYLS = 16) is a calculated value derived from:

```
total number of blocks / number of blocks per access position
```

For the control unit type and the device type, the last two digits of each are hex codes for the associated models. See the device documentation to determine the models for the hex codes given.

Response 9:

If you enter QUERY DASD DETAILS and the device has been expanded and has not yet been varied offline or online to make use of the expanded space, you see the following line in the response:

```
PENDING EXPANSION DETAILS: DEVTYPE = 3390-0A, CYLS = 47229
```

The DEVTYPE contains the model which might or might not have changed. The CYLS contains the new size which will be used by CP once the device is varied offline and online. If the DASD is an emulated device that represents a real SCSI device, the pending expansion information will not be included in the response. Use QUERY EDEVICE DETAILS to see the pending expansion information.

Response 10:

If you issue QUERY DASD RESERVE and the command completes successfully, you would receive the following response:

```
DASD rdev
```

If the command completed successfully but intervention is required, you would receive both of the following responses:

```
DASD rdev
```

and

```
DASD rdev
INT REQUIRED (if the device requires intervention)
DASD rdev <CP SYSTEM> nnnnn volid
      <CP OWNED >
<RESERVED BY      > <USER userid>
<RESERVE PENDING FOR> <SYSTEM      >
```

Response	Meaning
CP SYSTEM	indicates that this is a system DASD.
CP OWNED	indicates that this is a CP-owned volume.
<i>nnnnn</i>	specifies the number of links to this pack.
RESERVED BY	indicates that there is currently a reserve on the DASD.

**Response                    Meaning**

RESERVE PENDING FOR indicates that the user (or the system) is waiting for a reserve on the DASD.

USER *userid* indicates that the real reserve is held by the system for this user.

**Response 11:**

If you issued QUERY DASD RESERVE and no reserved or pending reserved DASD exists, you would receive the following response:

```
A reserved DASD was not found.
```

**Response 12:**

When you QUERY DASD SPACE-EFFICIENT:

You will receive the "not a space-efficient" response if the disk is not in a space-efficient configuration.

You will receive the "not available" message if the hardware indicates that it cannot currently produce the required information.

The pool data is the extent pool ID and the size of the extent pool repository.

The capacity data is the space-efficient volume capacity limit and the capacity limit warning watermark (%).

The guaranteed data is the space-efficient volume guaranteed capacity and the guaranteed capacity warning watermark (%).

The space data is the extent pool space currently allocated to this volume and the space-efficient volume capacity (logical volume size). The ratio of these two is computed as a percentage. If no space is currently defined, the display indicates a value of "- -". If the denominator is zero, - -% will be displayed.

```
q dasd space-efficient
DASD FCE0 RWSCE0 POOL 1 SIZE 1263286 CYLS
VOLUME CAPACITY LIMIT 0 CYLS, 0%
VOLUME GUARANTEED CAPACITY 0 CYLS, 0%
VOLUME SPACE 30 CYLS USED OF 3339 CYLS DEFINED, 1%
DASD FCE1 RWSCE0 POOL 1 SIZE 1263286 CYLS
VOLUME CAPACITY LIMIT 0 CYLS, 0%
VOLUME GUARANTEED CAPACITY 0 CYLS, 0%
VOLUME SPACE 0 CYLS USED OF 3339 CYLS DEFINED, 0%
DASD FEE0 RWSEE0 POOL 1 SIZE 1263286 CYLS
VOLUME CAPACITY LIMIT 0 CYLS, 0%
VOLUME GUARANTEED CAPACITY 0 CYLS, 0%
VOLUME SPACE 20 CYLS USED OF 3339 CYLS DEFINED, 1%
DASD FEE1 RWSEE1 POOL 1 SIZE 1263286 CYLS
VOLUME CAPACITY LIMIT 0 CYLS, 0%
VOLUME GUARANTEED CAPACITY 0 CYLS, 0%
VOLUME SPACE 20 CYLS USED OF 3339 CYLS DEFINED, 1%
Ready; T=0.01/0.01 16:25:14

q dasd space-efficient 1234
A Space-Efficient DASD was not found
Ready; T=0.01/0.01 16:27:40

q dasd space-efficient 1234
DASD 1234 MYDISK: Space-Efficient information is not available
Ready; T=0.01/0.01 16:27:40
```

**Response 13:**

If you issued a QUERY DASD DETAILS command to a volume that resides on a data encryption drive (DED), and/or a solid-state drive (SSD), the "DEVICE DETAILS" line of output will include indicators stating the presence of these facilities. The output will look similar to the following:

```
q dasd details 521d
521D CUTYPE = 2107-E8, DEVTYPE = 3390-0A, VOLSER = ERF001, CYLS = 3339
  CACHE DETAILS:  CACHE NVS CFW DFW PINNED CONCOPY
                   -SUBSYSTEM  Y  Y  Y  -  N  N
```

## QUERY DASD

```
-DEVICE Y - - Y N N
DEVICE DETAILS: CCA = 1D, DDC = --, DED = YES, SSD = NO, SS = 00,
ACTIVE = Y, ESE = N
DUPLEX DETAILS: --
CU DETAILS: SSID = 0102, CUNUM = 5200
SERIAL NUMBER: 01XX-12345
```

If QUERY DASD DETAILS is issued against an emulated device (EDEV) that includes one or both of these facilities, the "DEVICE DETAILS" line of output will be added to the response, but the CCA/DDC text will not be displayed.

```
q dasd details dead
DEAD CUTYPE = 6310-80, DEVTYPE = 9336-10, VOLSER = ERF105, CYLS = 10796
BLKS = 8388608
DEVICE DETAILS: DED = YES, SSD = NO, SS = 00
```

Regardless of device type, at least one facility must be present on a volume for the DED and SSD fields to be displayed.

Response 14:

If you issue a QUERY DASD DETAILS command to a real device that supports Soft Fence and Query Host Access, the soft fence state, CPU serial number, and LPAR ID of the Host System accessing the device will be shown as follows:

```
q dasd details ae50
AE50 CUTYPE = 2107-E8, DEVTYPE = 3390-0A, VOLSER = ERF001, CYLS = 3339
CACHE DETAILS: CACHE NVS CFW DFW PINNED CONCOPY
-SUBSYSTEM Y Y Y - N N
-DEVICE Y - - Y N N
DEVICE DETAILS: CCA = 00, DDC = --, SS = 0, ACTIVE = Y, ESE = N
DUPLEX DETAILS: --
CU DETAILS: SSID = 040A, CUNUM = AE50
SERIAL NUMBER: 01XX-12345
FENCED STATE: NONE
HOST ACCESS: CPU PARTITION GROUPED RESERVED MPATHMODE
312E 15 Y N Y
-LOCAL E276 0D Y N Y
```

The soft fence state will be shown in the response only when the control unit supports the Soft Fence feature. The same applies to Host Access information; that is, whether Host Access information will be shown also depends on the control unit support.

The Host Access section of the response contains the following columns:

### CPU

indicates the CPU serial number.

### PARTITION

indicates the LPAR number.

### GROUPED

indicates the grouped state of the device.

### RESERVED

indicates if the device is reserved to the path group.

### MPATHMODE

indicates if the device is in multi-path mode.

Within the Host Access section, Y and N indicate the following status:

Column	Y	N
GROUPED	The device is grouped to one or more paths in the path group.	The device is not grouped to any path in the path group.
RESERVED	The device is reserved to this path group.	The device is not reserved to this path group.

Column	Y	N
MPATHMODE	The device is grouped in multi-path mode.	The device is in single path mode.

**Note:**

- The CPU serial number and LPAR number come from the CPU identifier of the first-level system used to form the real Path Group ID. They cannot be changed by the SET CPUID command used to change the CPU identifier of a virtual machine. The LPAR number matches the partition ID that can be found in the activation profile for an LPAR.
- Host Access information is obtained only for dedicated DASD and fullpack minidisks, and all of the status information in this response shows what is out on the real hardware. For example, RESERVED shows the real reserved state for the specified device, not the virtual reserve.

Response 15:

If you issue a QUERY DASD DETAILS command to a real device, but errors occur during the processing of Soft Fence and Query Host Access related I/O that prevents the successful completion of the command, the following response is displayed:

```
q dasd details 521d
AE50 CUTYPE = 2107-E8, DEVTYPE = 3390-0A, VOLSER = ERF001, CYLS = 3339
CACHE DETAILS:  CACHE NVS  CFW  DFW  PINNED  CONCOPY
  -SUBSYSTEM   Y    Y    Y    -    N    N
  -DEVICE      Y    -    -    Y    N    N
DEVICE DETAILS: CCA = 00, DDC = --, SS = 0, ACTIVE = Y, ESE = N
DUPLEX DETAILS: --
CU DETAILS: SSID = 040A, CUNUM = AE50
SERIAL NUMBER: 01XX-12345
FENCED STATE: I/O Error
HOST ACCESS: I/O Error
```

Response 16:

If you issue a QUERY DASD DETAILS command and the device is in HyperSwap processing, Soft Fence and Query Host Access related I/O will be suppressed and the following response will be issued:

```
q dasd details 521d
AE50 CUTYPE = 2107-E8, DEVTYPE = 3390-0A, VOLSER = ERF001, CYLS = 3339
CACHE DETAILS:  CACHE NVS  CFW  DFW  PINNED  CONCOPY
  -SUBSYSTEM   Y    Y    Y    -    N    N
  -DEVICE      Y    -    -    Y    N    N
DEVICE DETAILS: CCA = 00, DDC = --, SS = 0, ACTIVE = Y, ESE = N
DUPLEX DETAILS: --
CU DETAILS: SSID = 040A, CUNUM = AE50
SERIAL NUMBER: 01XX-12345
FENCED STATE: SUPPRESSED BY HYPERSWAP
HOST ACCESS: SUPPRESSED BY HYPERSWAP
```

Response 17:

If you issue a QUERY DASD DETAILS command but the user has option QUICKDSP set on, I/O for the Soft Fence and Query Host Access functions will not be issued and the response is displayed as follows:

```
q dasd details 521d
AE50 CUTYPE = 2107-E8, DEVTYPE = 3390-0A, VOLSER = ERF001, CYLS = 3339
CACHE DETAILS:  CACHE NVS  CFW  DFW  PINNED  CONCOPY
  -SUBSYSTEM   Y    Y    Y    -    N    N
  -DEVICE      Y    -    -    Y    N    N
DEVICE DETAILS: CCA = 00, DDC = --, SS = 0, ACTIVE = Y, ESE = N
DUPLEX DETAILS: --
CU DETAILS: SSID = 040A, CUNUM = AE50
FENCED STATE: QUICKDSP option set
HOST ACCESS: QUICKDSP option set
```

Response 18:

If you issue a QUERY DASD DETAILS command but the device is a HyperPAV alias device, the I/O needed for the Query Host Access function will not be issued and the response is displayed as follows:

## QUERY DASD

```
q dasd details 521d
AE50 CUTYPE = 2107-E8, DEVTYPE = 3390-0A, VOLSER = ERF001, CYLS = 3339
CACHE DETAILS:  CACHE NVS CFW DFW PINNED CONCOPY
  -SUBSYSTEM   Y   Y   Y   -   N   N
  -DEVICE      Y   -   -   Y   N   N
DEVICE DETAILS: CCA = 00, DDC = --, SS = 0, ACTIVE = Y, ESE = N
DUPLICATE DETAILS: --
CU DETAILS: SSID = 040A, CUNUM = AE50
SERIAL NUMBER: 01XX-12345
FENCED STATE: NONE
HOST ACCESS: --
```

Response 19:

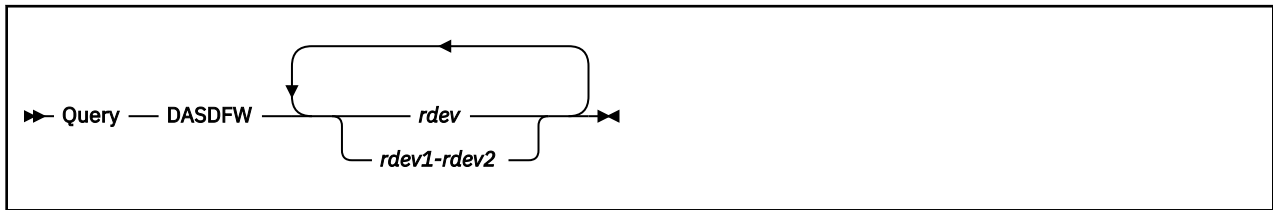
If you issue a QUERY DASD CHARACTERISTICS command, the following response is displayed:

```
q dasd char 521d
- Begin - RDEV 521D - Read Configuration Data -
DC010100 F0F0F2F1 F0F7F9F0 F0C9C2D4 F7F5F0F0 F0F0F0F0 F0E8F5F8 F1F1071D
D4020000 F0F0F2F1 F0F7F9F3 F2C9C2D4 F7F5F0F0 F0F0F0F0 F0E8F5F8 F1F10700
D0000000 F0F0F2F1 F0F7F9F3 F2C9C2D4 F7F5F0F0 F0F0F0F0 F0E8F5F8 F1F00700
F0000001 F0F0F2F1 F0F7F9F0 F0C9C2D4 F7F5F0F0 F0F0F0F0 F0E8F5F8 F1F10700
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
80000310 2D001E00 05070013 62131325 000CC01D 3905FADB 03100000 0000F200
- End - RDEV 521D - Read Configuration Data -
- Begin - RDEV 521D - Read Device Characteristics -
2107E833 900A5F8C 5FF72024 01F4000F E000E5A2 05940222 13090674 00000000
00000000 00000000 24241F02 DFEE0001 0677080F 007F4A00 003C0000 000001F4
- End - RDEV 521D - Read Device Characteristics -
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP006E Invalid device type - *rdev*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not specified or it is invalid.
- HCP040E Device *rdev* does not exist
- HCP263E Too many operands were supplied for this command.
- HCP332E Invalid control unit type - *rdev*
- HCP8712E Read Configuration Data failed on device *rdev*
- HCP8712E Read Device Characteristics failed on device *rdev*

## QUERY DASDFW



### Authorization

Privilege Class: B

### Purpose

Use the QUERY DASDFW command to display the DASD fast write status of devices attached to cache storage subsystems.

### Operands

*rdev*

*rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers of the devices being queried.

### Usage Notes

1. The QUERY DASDFW command is only valid for cache storage subsystems that support the DASD fast write function. An error message is generated if the specified subsystem does not support this function.
2. Caching for RAID (Redundant Array of Independent Disks) technology DASD such as an IBM DASD subsystem is always active. It cannot be modified by the user.
3. This command may be issued to offline devices.

### Responses

Response 1:

```

rdev DASD fast write function {is active.           }
                             {is inactive.         }
                             {destaging has failed.}
                             {is suspended.       }
  
```

This response is repeated for each specified device when you enter the QUERY DASDFW command.

*rdev*

is the real device number of a device being queried.

**is active**

indicates that the DASD fast write function is active for the device.

**is inactive**

indicates that the DASD fast write function is inactive for the device.

**destaging has failed**

indicates that deactivation of DASD fast write is pending because transfer of modified data to DASD has failed.

### **is suspended**

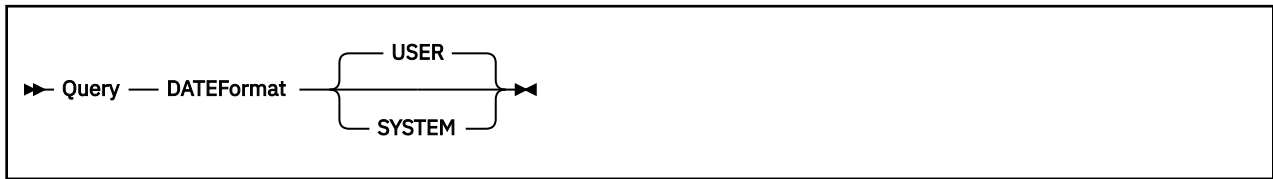
indicates that the DASD fast write function is suspended for the device. Suspension occurs when an authorized guest issues the Perform Subsystem Function channel command to suspend all fast write operations.

### **Messages**

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*



## QUERY DATEFORMAT



### Authorization

Privilege Class: G

### Purpose

Use QUERY DATEFORMAT to display the current default date format for the system or an individual user.

### Operands

#### USER

tells CP to display the default date format for this user.

#### SYSTEM

tells CP to display the default date format for the system.

### Usage Notes

1. If your user default date format is set to (or defaults to) the system-wide default date format, and the system-wide default is changed, the response to QUERY DATEFORMAT USER will continue to indicate the old system-wide default as your user default until you either log off and log back on or issue the SET DATEFORMAT SYSDEFAULT command to switch to the new system-wide default.

### Responses

Response 1:

If you issue QUERY DATEFORMAT USER, and your user default date format is set to SHORTDATE, FULLDATE, or ISODATE, you will get one of the following responses:

```
User Dateformat = SHORTDATE
```

```
User Dateformat = FULLDATE
```

```
User Dateformat = ISODATE
```

Response 2:

If you issue QUERY DATEFORMAT USER, and your user default date format is set to (or defaults to) the system-wide default, you will get one of the following responses:

```
User Dateformat = SHORTDATE (SYSDEFAULT)
```

```
User Dateformat = FULLDATE (SYSDEFAULT)
```

```
User Dateformat = ISODATE (SYSDEFAULT)
```

Response 3:

If you issue QUERY DATEFORMAT SYSTEM, you will get one of the following responses:

```
System Dateformat = SHORTDATE
```

## QUERY DATEFORMAT

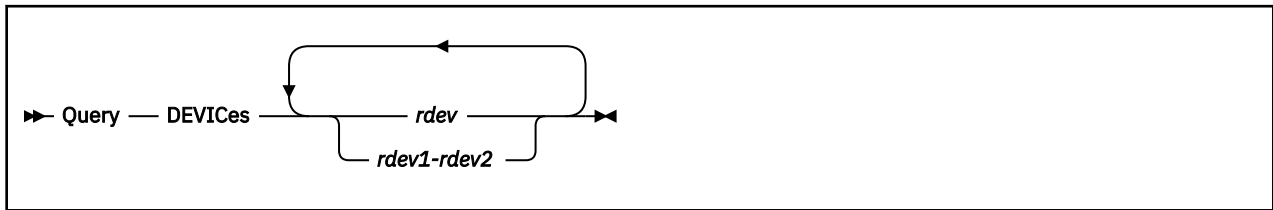
System Dateformat = FULLDATE

System Dateformat = ISODATE

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - Invalid option - command contains extra option(s) starting with *option*

## QUERY DEVICES



### Authorization

Privilege Class: B

### Purpose

Use the QUERY DEVICES command to display the settings that CP will use if a device is added dynamically to the I/O configuration or is powered on after IPL.

### Operands

#### *rdev*

is the device number of a real device.

#### *rdev1-rdev2*

is a range of real device numbers.

### Usage notes

- Device numbers in the range 0000-FFFF are valid.
- The same settings are used for a device number in all subchannel sets. For example, the settings for device number 1234 are also used for device numbers 11234, 21234, and 31234.
- The initial settings are established by the DEVICES statement in the system configuration file. After IPL, you can change the settings by using the SET DEVICES and SET RDEVICE commands. Those commands affect CP's handling of a new device only when the commands are issued at the following times:
  - Before a device is detected due to a dynamic I/O change that adds the device to the LPAR.
  - Before you power on a device that was unavailable at IPL.

The SET DEVICES command can change the settings that the QUERY DEVICES command displays. When the SET RDEVICE command is issued for a device after the device is initialized, the setting does not affect the settings that the QUERY DEVICES command displays.

- After a device is initialized, you can change some of its settings by using the SET RDEVICE, SET SCMEASURED, and SET THROTTLE commands. The changes are not displayed by QUERY DEVICES and do not affect the settings that CP will use if a device is added dynamically to the I/O configuration or is powered on after IPL.

### Examples

Example 1:

To find out how CP handles device 2345 and devices in range 3401-3420, enter the following command:

```
query devices 2345 3401-3420
```

If the command is successful, you get a response that is like the following example:

## QUERY DEVICES

Device(s)	Accepted	Online at IPL	Sensed	Sensed /Offln	Shared	Dynam I/O	SubCh Meas	Throttled	Assign at IPL
2345	Y	N	Y	N	Y	Y	Y	Y	N
3401-3410	Y	Y	Y	N	N	N	Y	Y	Y
3411-3420	Y	Y	Y	N	N	N	Y	N	Y

### Responses

The response line follows the headings line and is repeated for each device number or range of device numbers that have the same settings. The response line indicates Y (yes) or N (no) for each setting. The response reports the following settings:

#### Accepted

indicates whether CP accepts the device when the device is dynamically added to VM from another partition.

#### Online at IPL

indicates whether CP initializes a device.

#### Sensed

indicates whether CP uses the information that is associated with a sense ID request to establish the device class or type.

#### Sensed /Offln

indicates whether CP does not initialize the device but still issues a sense ID request to determine the device class and type.

#### Shared

indicates whether CP can share the DASD device with other operating systems.

#### Dynam I/O

indicates whether CP allows dynamic I/O changes for the device.

#### SubCh Meas

indicates whether CP collects subchannel measurement data for the device.

#### Throttled

indicates whether CP limits (throttles) the rate of I/O coming from the device.

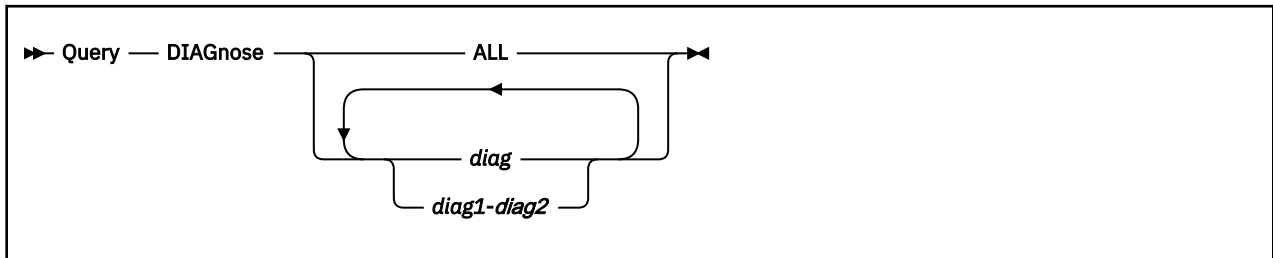
#### Assign at IPL

indicates whether CP assigns the tape drive device to the system when the device is brought online.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6546E HCD currently controls the software configuration. All CP software configuration commands are disabled.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*
- HCP6706E Invalid string - *string*
- HCP6709E Too many items specified - *item*

## QUERY DIAGNOSE



### Authorization

Privilege Class: A, C, E

### Purpose

Use QUERY DIAGNOSE to display information (status, entry point names, and privilege classes) about one or more DIAGNOSE codes.

### Operands

#### ALL

tells CP to display information about all existing DIAGNOSE codes.

#### *diag*

#### *diag1-dia2*

is the number of the DIAGNOSE code for which you want information. Each *diag* must be a hexadecimal number between X'0000' and X'03FC'. You can specify a single DIAGNOSE code, a range of DIAGNOSE codes, or any combination thereof.

### Usage Notes

1. If you do not specify the ENABLE operand on the DEFINE DIAGNOSE command, a new DIAGNOSE code is initially in a disabled state after being defined. CP treats disabled DIAGNOSE codes as if they were never defined. If you try to use a disabled DIAGNOSE code in a program, CP will give you a program check specification exception.
2. To activate a new DIAGNOSE code:
  - While defining it, use the ENABLE operand of the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE”](#) on page 172).
  - After defining it, use the ENABLE DIAGNOSE command (see [“ENABLE DIAGNOSE”](#) on page 463).
3. To change the definition of an existing DIAGNOSE code after initialization, use the MODIFY DIAGNOSE command (see [“MODIFY DIAGNOSE”](#) on page 697).
4. To display the address of the CP DIAGNOSE code table block for a DIAGNOSE code, use the LOCATE DGNBK command (see [“LOCATE DGNBK”](#) on page 608).
5. To deactivate a DIAGNOSE code:
  - While defining it, use the DISABLE operand of the DEFINE DIAGNOSE command (see [“DEFINE DIAGNOSE”](#) on page 172).
  - After defining it, use the DISABLE DIAGNOSE command (see [“DISABLE DIAGNOSE”](#) on page 316).
6. Many external security managers (ESMs) do not support DIAGNOSE codes above X'03FC'. For this reason, CP does not support DIAGNOSE codes above X'03FC'. The DIAGNOSE codes between X'0000' and X'03FC' are divided as follows:

## QUERY DIAGNOSE

### **X'0000' to X'00FC'**

Reserved for IBM use

### **X'0100' to X'01FC'**

Reserved for customer use

### **X'0200' to X'03FC'**

Reserved for IBM use.

7. For more information about user-defined DIAGNOSE codes, see [Defining and Modifying Commands and Diagnose Codes](#) in *z/VM: CP Exit Customization*.

## Responses

Response 1:

To display the current information for DIAGNOSE code X'28' and DIAGNOSE codes X'30' through X'3C', enter the following:

```
query diagnose 28 30-3f
Code Status EPName PrivClasses
0028 Enabled HCPDCHNM <ANY>
0034 Enabled HCPHVC34 CE
003C Enabled HCPUDDS ABC
Ready;
```

This response tells us that DIAGNOSE codes X'28', X'34', and X'3C' are enabled and that DIAGNOSE codes X'30' and X'38' are not defined.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP021E A Diagnose code was not supplied or it is not valid
- HCP2751E Diagnose code *diag* does not exist.
- HCP2751E Diagnose codes *diag1-dia2* do not exist.
- HCP6706E Invalid Diagnose code - *diag*

# QUERY DUMP

► Query — DUMP ◄

## Authorization

Privilege Class: B

## Purpose

Use QUERY DUMP to display the following:

- The device type and unit or units assigned to receive CP abnormal termination dumps
- The current settings of the DUMP options.

## Usage Notes

1. For DASD only, if you have identified more than one device to receive CP dumps or CP has obtained DASD dump space from more than one device during system initialization and operation, CP responds with the device type DASD, the real device numbers, and the pages allocated from each.
2. If the system was unable to find sufficient SPOOL space, it responds with 'No dump unit'. 'No dump unit' indicates that, at the time CP is ready to create the dump, no dump will be created. CP will still create a symptom record to log the occurrence of the event.

## Responses

Response 1:

When one CP-owned DASD has been assigned with system abend dump space, the response is as follows:

```
{type rdev} dump unit CP {IPL } [pages nnnnnnnn] PGMBKS {DEFAULT } FRMTBL {DEFAULT}
                    {NOIPL}                    {ALL } {YES }
                    {NONE }                    {NO }
                    {FAILUSER}
```

### **type rdev**

indicates the device type and the real device number of the dump device.

### **CP**

is included for compatibility purposes.

### **IPL**

### **NOIPL**

indicates whether the system automatically restarts when the dump is completed. NOIPL indicates that the system enters a wait state upon completion of the dump.

### **pages nnnnnnnn**

indicates the number of pages of spool space, by device, allocated for system dumps. *nnnnnnnn* is a 1-to 8-digit decimal number.

### **FRMTBL DEFAULT|YES|NO**

indicates whether the full CP frame table is included in a hard abend dump. If DEFAULT is displayed, the frame table is included in the cases for which IBM believes the frame table is needed to debug the problem. Otherwise, the dump includes a correlation table describing only the frames whose contents were dumped. YES forces inclusion of the frame table and NO forces the dump to have a correlation table. IBM recommends using the DEFAULT setting unless IBM support personnel instruct you to do otherwise in order to diagnose a specific problem.

**PGMBKs DEFAULT|ALL|NONE|FAILUSER**

indicates whether user PGMBKs are included in a hard abend dump. If DEFAULT is displayed, PGMBKs are included in the cases for which IBM believes they are needed to debug the problem. ALL forces inclusion of all PGMBKs that aren't paged out, and NONE omits user PGMBKs. FAILUSER causes:

- PGMBKs to be included for the guest that was running on the failing processor at the time of the abend.
- PGMBKs owned by the system to be included.

IBM recommends using the DEFAULT setting unless IBM support personnel instruct you to do otherwise in order to diagnose a specific problem.

The frame table and PGMBKs together constitute the vast majority of CP-owned frames, because they contain entries describing each 4K unit of real and virtual memory, respectively. The CP frame table is 1/128th the size of configured memory (8 GB per TB). PGMBKs take up 8K for each 1MB of virtual memory; though they are pageable, a PGMBK must remain resident as long as any page of that megabyte is resident. Omitting these structures from dumps when they are not likely to be needed for diagnosis dramatically reduces dump size and dumping time. With the DEFAULT operand, CP determines whether or not to include the structures based on the abend code and failing module.

Response 2:

When more than one CP-owned DASD has been assigned with system abend dump space, the response is as follows:

Line 1:

```
DASD DUMP SPACE  {CP }  {IPL } PGMBKs {DEFAULT } FRMTBL {DEFAULT}
                  {NOIPL}  {ALL }  {YES }
                  {NONE }  {NO  }
                  {FAILUSER}
```

Lines 2 and 3:

```
RDEV  PAGES
----  -
```

Lines 4 and following:

```
rdev  nnnnnnnnn
:
```

CP, IPL/NOIPL, PGMBKs, and FRMTBL are all explained under Response 1.

Response 3:

When the SET DUMP OFF command has been issued and accepted, the response is:

```
No dump unit - dump function is SET OFF
```

Response 4:

```
No dump unit - insufficient spool space
```

indicates that the system was unable to find sufficient spool space for a system abend dump on CP-owned system DASD at system initialization or after a non-destructive dump (snapdump). This means that at the time CP is ready to create the dump, no dump will be created. In this situation, CP will just create a symptom record when the dump would have occurred.

The system operator should re-issue the SET DUMP command. If a DASD dump is desired, create enough empty spool space to enable the dump to be allocated.



**Messages**

- HCP002E Invalid operand - *operand*
- HCP9259I No dump unit is set. No system abend dump will be generated. To assign a device, enter the SET DUMP command.

## QUERY DUMPDEV



### Authorization

Privilege Class: G

### Purpose

Use the **QUERY DUMPDEV** command to display DUMPDEV parameters for a guest IPL. The parameters are for a list-directed IPL with dump from a SCSI or ECKD device. The parameters can identify the system dumper program and information to pass to the program. The parameters can also identify the virtual device number and specify whether the IPL is a secure IPL. The device that is IPLed is also used to store the dump data.

The DUMPDEV parameters can be set by using the **SET DUMPDEV** command.

For more information, see [SET DUMPDEV](#) in *z/VM: CP Commands and Utilities Reference*.

### Responses

The following lines are displayed if DUMPDEV parameters are set for a SCSI device:

```
SCSI      DEVICE {TBD } {NOSECURE}
           {ddd} {SECURE }
PORTNAME hhhhhhhh hhhhhhhh LUN hhhhhhhh hhhhhhhh BOOTPROG {n }
           {AUTO}
BR_LBA   hhhhhhhh hhhhhhhh
```

The following lines are displayed if DUMPDEV parameters are set for an ECKD device:

```
ECKD      DEVICE {TBD } {NOSECURE}
           {ddd} {SECURE }
BOOTPROG {n } BOOTREC {cylinder head record}
           {AUTO} {LABEL }
```

The following lines are displayed if hex SCPDATA is set:

```
SCPDATA
 00----03 04----07 08----11 12----15 16----19
0000 hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
0020 hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
0040 hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
.
.
```

The following lines are displayed if text SCPDATA is set:

```
SCPDATA
 0-----+----1-----+----2-----+----3-----+----4-----+----
0000 ttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
0050 ttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
0100 ttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
.
.
```

**SCSI**

The SCSI response indicates that DUMPDEV parameters are set for the IPL of a SCSI device. SCSI is the default device type.

**ECKD**

The ECKD response indicates that DUMPDEV parameters are set for the IPL of an ECKD device.

**DEVICE TBD****DEVICE *dddd***

The DEVICE TBD response indicates that the DUMPDEV parameters do not include the device number. The device number is to be specified later.

If the SECURE parameter is not set, then the device number can be specified later as an operand of the **IPL** command. If the SECURE parameter is set, then the device number must be set by using the **SET DUMPDEV** command.

The DEVICE *dddd* response indicates the 4-digit hexadecimal virtual device number (*dddd*) of the device to be IPLed.

There is no default value for the DEVICE parameter.

**PORTNAME *hhhhhhhh hhhhhhhh***

The PORTNAME response indicates the hexadecimal 8-byte fibre channel port name of the SCSI device. The default value is 0.

**LUN *hhhhhhhh hhhhhhhh***

The LUN response indicates the hexadecimal 8-byte logical unit number of the SCSI device. The default value is 0.

**BOOTPROG *n*****BOOTPROG AUTO**

The BOOTPROG response indicates the boot program that is loaded from the IPL device. The *n* value indicates the program number. The program number must be a decimal value in the range 0 - 30. The BOOTPROG AUTO response indicates that the first dump program (not operating system) on the IPL device is loaded. The default value is BOOTPROG 0.

**BR\_LBA *hhhhhhhh hhhhhhhh***

The BR\_LBA response indicates the hexadecimal logical-block address of the boot record of the SCSI device. The default value is 0.

**BOOTREC *cylinder head record*****BOOTREC LABEL**

The BOOTREC response indicates the location of the boot record. The location can be a specification of cylinder, head, and record on the IPL device. The values are delimited by blanks and contain no leading zeros. The values must be in hexadecimal format.

A BOOTREC LABEL response indicates that the location of the boot record is determined from the volume label. BOOTREC LABEL is the default setting.

**SECURE****NOSECURE**

The SECURE response indicates that the SECURE parameter is set. The NOSECURE response indicates that the SECURE parameter is not set. The SECURE parameter must be set before you can initiate a secure IPL. NOSECURE is the default setting.

**SCPDATA**

The SCPDATA response indicates the information that is passed to the IPL program. The information is displayed in the same format (text or hex) as it was entered.

**Usage Notes**

- Parameters for a list-directed IPL without dump (also called LOADDEV parameters) are set by using any of the following methods:
  - **SET LOADDEV** command.

## QUERY DUMPDEV

- LOADDEV user directory statement. The LOADDEV user directory statement can be modified by a directory management tool such as IBM Directory Maintenance (DirMaint). To set load parameters by using DirMaint, use the **DIRM LOADDEV** command. The LOADDEV user directory statement can be modified also by the `Image_IPL_Characteristics_Define_DM` API function.

Parameters for a list-directed IPL **with** dump (also called DUMPDEV parameters) are set by using the CP **SET DUMPDEV** command.

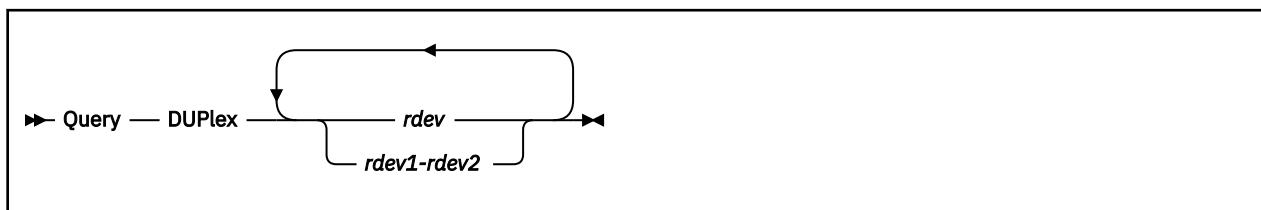
For more information, see [SET LOADDEV](#) and [SET DUMPDEV, LOADDEV Directory Statement](#) in *z/VM: CP Planning and Administration*, [DIRM LOADDEV](#) in *z/VM: Directory Maintenance Facility Commands Reference*, and [Image\\_IPL\\_Characteristics\\_Define\\_DM](#) API function in *z/VM: Systems Management Application Programming*.

2. If a DUMPDEV parameter is not set and has a default value, then the **QUERY DUMPDEV** response displays the default parameter value.
3. The default LOADDEV and DUMPDEV values for the PORTNAME and LUN parameters is 0, which is not a valid device port name and not a valid device LUN. An IPL attempt that uses the default value for the PORTNAME or LUN parameter will fail.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP2824I No DUMPDEV parameters are currently defined

## QUERY DUPLEX



### Authorization

Privilege Class: B

### Purpose

Use the QUERY DUPLEX command to display the status of duplex pairs of devices attached to the 3990-3 and 3990-6 storage subsystem.

### Operands

**rdev**

**rdev1-rdev2**

is the real device number, a list of real device numbers, or a range of real device numbers of the primary or secondary volumes in the established duplex pairs.

### Usage Notes

1. The device number of either the primary or secondary volume in a duplex pair may be specified on this command.
2. This command may be issued to offline devices.
3. If both the primary and secondary of a duplex pair are included in the device list or device range, two identical responses will be displayed.
4. A VIRTUAL version of this command, QUERY VIRTUAL DUPLEX, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL DUPLEX”](#) on page 1295.

### Responses

Response 1:

```
Duplex pair (prdev,srdev) - active
```

This response is displayed for each duplex pair that is not in suspended duplex state and for which synchronization is not in progress.

When CP is running in a guest virtual machine, hyphens (----) will appear in the *srdev* position if the secondary volume is not in the guest's CP I/O configuration.

**prdev**

is the real device number of the primary volume in the duplex pair.

**srdev**

is the real device number of the secondary volume in the duplex pair.

Response 2:

## QUERY DUPLEX

```
Duplex pair (prdev,srdev) - copy operation  
is in progress.
```

This response is displayed for each duplex pair that is in the process of being synchronized.

When CP is running in a guest virtual machine, hyphens (----) will appear in the *srdev* position if the secondary volume is not in the guest's CP I/O configuration.

### ***prdev***

is the real device number of the primary volume in the duplex pair.

### ***srdev***

is the real device number of the secondary volume in the duplex pair.

Response 3:

```
Duplex pair (prdev,srdev) - suspended
```

This response is displayed for each duplex pair that is in the *suspended duplex* state. Either the primary or the secondary volume has been suspended.

When CP is running in a guest virtual machine, hyphens (----) will appear in the *srdev* position if the secondary volume is not in the guest's CP I/O configuration.

### ***prdev***

is the real device number of the primary volume in the duplex pair.

### ***srdev***

is the real device number of the secondary volume in the duplex pair.

Response 4:

```
rdev The associated {primary|secondary}  
volume in the duplex pair is not online to VM.
```

This response is displayed when the device number of the primary or secondary volume is specified on the command and the device number of the other duplex volume cannot be located.

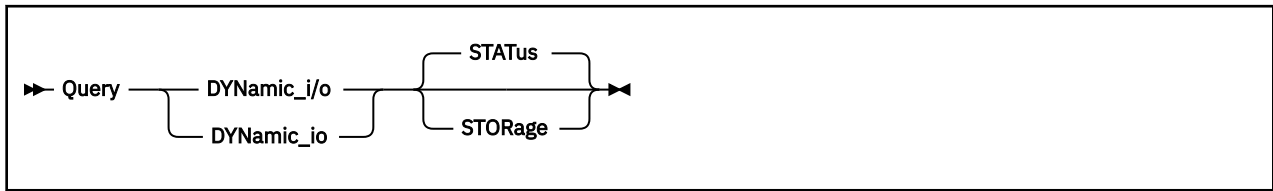
### ***rdev***

is the real device number of the primary or secondary volume in the duplex pair.

## Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP009E Invalid range - *range*
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*
- HCP1110E Device {*rdev/vdev*} is not in a duplex pair.

## QUERY DYNAMIC\_I/O



### Authorization

Privilege Class: B, E

### Purpose

Use QUERY DYNAMIC\_I/O or QUERY DYNAMIC\_IO to display:

- Whether the processor on which z/VM is running can perform dynamic I/O configuration of the channel subsystem, and if so, whether dynamic I/O changes are currently allowed.
- The approximate number of subchannels, control units, and devices you can dynamically add to your system before running out of hardware system area (HSA) space.

### Operands

#### STATus

tells CP to display whether the dynamic I/O support is installed on your processor, and if so, whether the dynamic I/O support is enabled on your z/VM system. If omitted, STATUS is the default.

#### STORAge

tells CP to display the approximate number of subchannels, control units, and devices you can dynamically add to your system before running out of hardware system area (HSA) space.

### Usage Notes

1. For more information about displaying status or storage information for dynamic I/O changes, see [Using Dynamic I/O Configuration in z/VM: I/O Configuration](#).
2. To enable or disable all dynamic I/O changes on your system:
  - After initialization, use the SET DYNAMIC\_I/O command (see [“SET DYNAMIC\\_I/O” on page 1502](#)).
  - During initialization, use the ENABLE or DISABLE DYNAMIC\_I/O operands of the FEATURES statement in your system configuration file. For more information about the [FEATURES Statement](#), see [z/VM: CP Planning and Administration](#).
3. To prevent anyone from issuing SET DYNAMIC\_I/O commands after initialization, use the ENABLE or DISABLE SET\_DYNAMIC\_I/O operands of the FEATURES statement in your system configuration file. For more information about the FEATURES statement, see the [FEATURES Statement in z/VM: CP Planning and Administration](#).
4. To enable or disable dynamic I/O changes for specific devices on your system:
  - After initialization, use the SET DEVICES command (see [“SET DEVICES” on page 1490](#)).
  - During initialization, use the DYNAMIC\_I/O or NOTDYNAMIC\_I/O operands of the DEVICES statement in your system configuration file. For more information about the [DEVICES Statement](#), see [z/VM: CP Planning and Administration](#).
5. To prevent anyone from issuing SET DEVICES commands after initialization, use the ENABLE or DISABLE SET\_DEVICES operands of the FEATURES statement in your system configuration file. For more information about the FEATURES statement, see the [FEATURES Statement in z/VM: CP Planning and Administration](#).

## Responses

Response 1 (Installed, Enabled):

If someone installed and enabled dynamic I/O support on your processor, you would see:

```
query dynamic_i/o status
Dynamic I/O changes are allowed on this system.
Ready;
```

Response 2 (Installed, Not Enabled):

If someone installed dynamic I/O support on your processor, but did not enable that support, you would see:

```
query dynamic_i/o status
No dynamic I/O changes are allowed on this system.
Ready;
```

Response 3 (Not Installed):

If no one installed dynamic I/O support on your processor, you would see:

```
query dynamic_i/o status
Dynamic I/O configuration is not supported.
Ready;
```

Response 4 (Installed, Enabled, Not Valid):

If someone installed and enabled dynamic I/O support on your processor, but your existing I/O configuration is not valid, you would see:

```
query dynamic_i/o status
Dynamic I/O changes are allowed but the configuration is invalid.
Ready;
```

When your system enters configuration mode (either explicitly, by entering the SET CONFIGMODE ON command, or automatically, by entering a dynamic I/O command), CP turns the "valid bit" off in the channel subsystem. If your dynamic I/O changes complete successfully, CP turns the valid bit back on before exiting configuration mode.

If one of the following events occur while you are in configuration mode:

- The system abends,
- You issue a LOGOFF command,
- You issue a SHUTDOWN REIPL command,
- You issue a CLEAR RESET from the hardware console, or
- You issue a LOAD CLEAR from the hardware console,

Your z/VM system or your hardware takes you out of configuration mode before shutting down the system, but they do not turn the valid bit back on. Thus, if you are interrupted while making dynamic I/O changes, your I/O configuration is not valid. To correct this situation, use the RESET operand of the SET CONFIGMODE command (see [“SET CONFIGMODE”](#) on page 1436).

If the system loses power while you are in configuration mode, your configuration will be valid when you bring it back up, because you choose which I/O configuration data set (IOCDs) file to use before performing the initial machine load (IML) or power-on reset (POR) to bring your system up.

Response 5 (Installed, Not Enabled, Not Valid):

If someone did not enable dynamic I/O support on your processor, and your existing I/O configuration is not valid, you would see:



```
query dynamic_i/o status
No dynamic I/O changes are allowed and the configuration is invalid.
Ready;
```

#### Response 6 (Non-LCSS System):

To display the number of subchannels, control units, and devices you can dynamically add to your system when there is no logical channel-subsystem facility installed, enter the following:

```
query dynamic_i/o storage
Remaining channel subsystem resources for configuration changes:
Number of subchannels for unshared paths: 649
Number of control units: 42
Number of unshared channel paths: 25
Number of subchannels for shared paths: 72
Number of shared channel paths: 3
Ready;
```

This response tells you that you can add a maximum of 649 subchannels on channel paths that are not shared, 72 subchannels on channel paths that are shared, 42 control units, 25 channel paths that are not shared, and 3 channel paths that are shared.

#### Response 7 (LCSS System):

To display the number of subchannels, control units, and devices you can dynamically add to your system when the logical channel-subsystem facility is installed, enter the following:

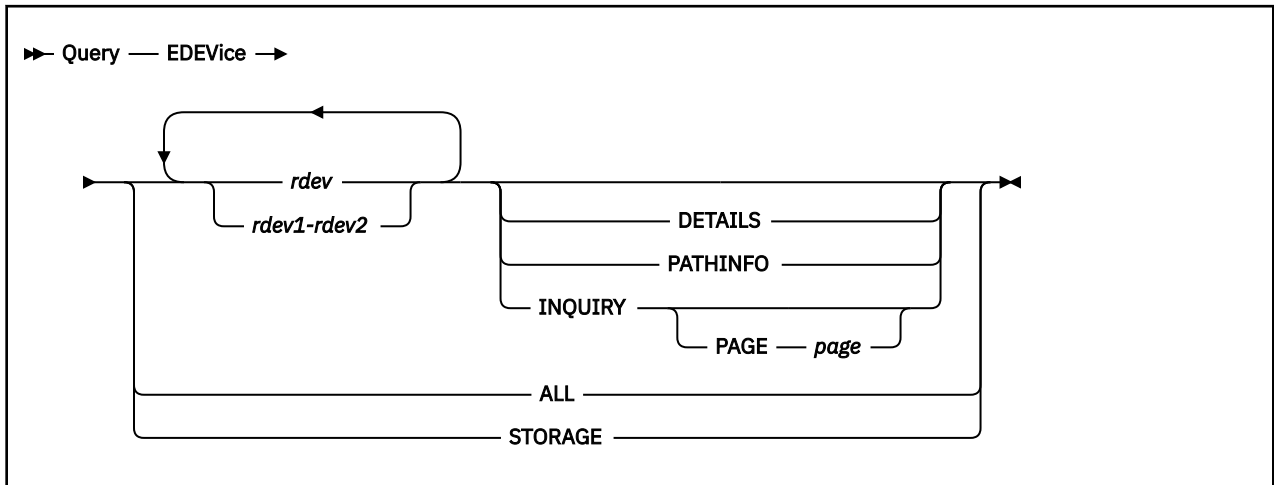
```
query dynamic_i/o storage
Remaining channel subsystem resources for configuration changes:
Number of control units: 63
Number of subchannel elements for CSS 0: 121
Number of channel paths for CSS 0: 13
Number of subchannel elements for CSS 1: 241
Number of channel paths for CSS 1: 21
Number of subchannel elements for CSS 2: 332
Number of channel paths for CSS 2: 32
Ready;
```

This example assumes 3 channel-subsystem images and the response tells you that you can add a maximum of 63 control units, 121 subchannels on CSS 0, 13 channel paths on CSS 0, 241 subchannels on CSS 1, 21 channel paths on CSS 1, 332 subchannels on CSS 2, and 32 channel paths on CSS 2.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*
- HCP6815E Command failed in channel subsystem.
- HCP6816E Dynamic I/O changes are being controlled by HCD.

## QUERY EDEVICE



### Authorization

Privilege Class: B

### Purpose

Use QUERY EDEVICE to display the device type and other detailed information associated with an emulated device.

### Operands

**rdev**

**rdev1-rdev2**

specifies the emulated devices for which you want information to be displayed. An *rdev* in the active configuration can be entered as a 4-digit hexadecimal device number between X'0000' and X'FFFF'. For an *rdev* in the active or standby configuration, the device number can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set of the device for which you want information to be displayed. A range of devices cannot span subchannel sets.

**DETAILS**

displays detailed information associated with an emulated device.

**ALL**

displays information for all emulated devices in the active configuration, but does not include the information that is displayed with the DETAILS operand.

**PATHINFO**

displays information associated with individual path activity.

**INQUIRY**

displays information from the standard inquiry page of the LUN associated with the emulated device.

The data returned by the INQUIRY option is in hexadecimal format. The IOEXPLOR EXEC can be used as an alternative in order to receive formatted data.

For further details on the information contained in the output, consult your storage hardware vendor.

**PAGE *page***

displays information from the specified vital product data page of the device. The *page* operand must be a hexadecimal page number between X'00' and X'FF'.

**STORAGE**

displays information for the pools of storage used by emulated devices, instead of the configuration of specific emulated devices. This operand will provide (in bytes) the configured size, amount used, and maximum amount used of each pool. It will also include the number of allocation and release requests that were satisfied by the pools.

**Usage Notes**

1. For queries that contain device numbers that represent emulated and non-emulated devices, responses will be generated only for emulated devices.
2. If you query an emulated device that was cleared with the SET EDEVICE command, the response will indicate that the device is still an emulated device, but no other information is associated with it.
3. Detailed information for emulated devices associated with 1750 and 2145 attributes includes an indication for each path specifying whether the path is a preferred (PREF) path or a not-preferred path (NOTPREF).
4. Some of the detailed information will be displayed only after the device is varied online successfully. If you do the QUERY EDEVICE DETAILS before the VARY ON or after a VARY OFFLINE, fewer details will be displayed.
5. The majority of the extra details information is obtained at VARY ON time, not when you issue the QUERY EDEVICE DETAILS command.
6. The maximum length of the inquiry response data display is 255 bytes.
7. An NVMe EDEVICE might indicate that it is not operational by displaying Int Req following the device attributes. The message indicates that intervention is required because there is a problem with the underlying NVMe PCI function. For example, the NVMe PCI function might have been removed from the configuration. To recover, the EDEVICES that are associated with the NVMe function must be deleted and redefined.

**Responses**

Response 1:

To query the emulated devices in the device range 300-305, enter the following:

```
q edev 300-305
EDEV 0300 TYPE FBA ATTRIBUTES 2105
EDEV 0301 TYPE FBA ATTRIBUTES SCSI
EDEV 0302 TYPE FBA ATTRIBUTES 2107
EDEV 0303 TYPE FBA ATTRIBUTES 1750
EDEV 0304 TYPE FBA ATTRIBUTES FLASH
EDEV 0305 TYPE FBA ATTRIBUTES NVM
Ready;
```

**Note:** Devices in the range that are not emulated devices do not appear in the response.

Response 2:

If none of the devices specified on the command are emulated devices, the following response will be displayed:

```
q edev 306-310
None of the devices specified are emulated devices.
Ready;
```

Response 3:

To query detailed information about an emulated device, for example, device 1111, enter the following:

```
q edev 1111 details
EDEV 1111 TYPE FBA ATTRIBUTES 2105
VENDOR: IBM          PRODUCT: 2105F20          REVISION: .293
BLOCKSIZE:          512  NUMBER OF BLOCKS: 390656
PATHS:
FCP_DEV: B908  WWPN: 5005076300CD04DA  LUN: 5144000000000000
```

## QUERY EDEVICE

```
CONNECTION TYPE: POINT_TO_POINT STATUS: ONLINE
EQID: ABCDEFGH
SERIAL NUMBER: 2146561344562
```

This response shows one possible value for STATUS: ONLINE. The other possible values are OFFLINE and RECOVERY. RECOVERY indicates that the system is undergoing error recovery and should return to ONLINE status soon.

Response 4:

To query detailed information about emulated devices in the device range 800-801, 850, and 888, enter the following:

```
q edev 800-801 details
EDEV 0800 TYPE FBA ATTRIBUTES 2105
VENDOR: IBM      PRODUCT: 2105F20      REVISION: .293
BLOCKSIZE:      512  NUMBER OF BLOCKS: 390656
PATHS:
  FCP_DEV: B046 WWPN: 5005076300C904DA LUN: 513A000000000000
  CONNECTION TYPE: SWITCHED STATUS: ONLINE
  FCP_DEV: B906 WWPN: 5005076300C904DA LUN: 513A000000000000
  CONNECTION TYPE: SWITCHED STATUS: ONLINE
EQID: IBM000002105000000000000000144132F6F200000000000000000
SERIAL NUMBER: 00112233445566
EDEV 0801 TYPE FBA ATTRIBUTES 2105
PATHS:
  FCP_DEV: B046 WWPN: 5005076300C904DA LUN: 513B00000000000000
EQID: ABCDEFGH

q edev 850 details
EDEV 0850 TYPE FBA ATTRIBUTES 1750
VENDOR: IBM      PRODUCT: 1750500      REVISION: 1.67
BLOCKSIZE:      512  NUMBER OF BLOCKS: 65536
PATHS:
  FCP_DEV: 2406 WWPN: 500507630E85FC66 LUN: 4010401200000000 NOTPREF
  CONNECTION TYPE: SWITCHED STATUS: ONLINE
  FCP_DEV: 2406 WWPN: 505507630E05FC66 LUN: 4010401200000000 PREF
  CONNECTION TYPE: SWITCHED STATUS: ONLINE
EQID: IBM000002105000000000000000144235F6F20000000000000FFFF
SERIAL NUMBER: 11223344556677

q edev 888 details
EDEV 0888 TYPE FBA ATTRIBUTES 2145
VENDOR: IBM      PRODUCT: 2145      REVISION: 0000
BLOCKSIZE:      512  NUMBER OF BLOCKS: 10485760
PATHS:
  FCP_DEV: 2FCB WWPN: 5005076801402797 LUN: 000E000000000000 NOT PREF
  CONNECTION TYPE: SWITCHED STATUS: ONLINE
  FCP_DEV: 2FCB WWPN: 5005076801402991 LUN: 000E000000000000 PREF
  CONNECTION TYPE: SWITCHED STATUS: ONLINE
EQID: IBM000002105000000000000000141462F6F20000000000000BB00
SERIAL NUMBER: 11223344556677
```

Note that the 801 device does not list much detail because it is not varied online.

Response 5:

To query detailed information about emulated device 500 (which has been cleared using the SET EDEVICE command), enter the following:

```
q edev 500 details
EDEV 0500 TYPE UNK
Ready;
```

Response 6:

If you enter QUERY EDEVICE DETAILS and the emulated device has been expanded and has not yet been varied offline and online to make use of the expanded space, you see the following line in the response:

```
PENDING EXPANSION NUMBER OF BLOCKS: 148493740
```

The blocks are the new size which will be used by CP once the device is varied offline and online.

Response 7:

To query information about the pools of reserved storage that are used for the configuration of emulated devices, enter the following:

```
q edev storage
```

	Size	Used	Maximum	Allocations		Releases	
				Total	Failed	Total	Failed
Primary:	122880	20928	21024	66	0	0	0
Secondary:	6524928	0	0	0	0	0	0

Ready; T=0.01/0.01 15:06:03

Response 8:

To query whether a real device at address 7F00 has an equivalency ID (EQID) assigned to it, enter the following:

```
q edev 7f00 details
```

```
11:58:21 EDEV 7F00 TYPE FBA ATTRIBUTES FLASH
11:58:21 VENDOR: IBM PRODUCT: FlashSystem-9840 REVISION: 1136
11:58:21 BLOCKSIZE: 512 NUMBER OF BLOCKS: 20971520
11:58:21 PATHS:
11:58:21 FCP_DEV: 1741 WWPN: 500507605E819CC1 LUN: 0000000000000000
11:58:21 CONNECTION TYPE: SWITCHED STATUS: ONLINE
11:58:21 EQID: 00020E662C220E6-0000-0201-0005F8C500000000013FFFFFF
11:58:21 SERIAL NUMBER: 20e662c220e6-0000-0166-0001a2
Ready; T=0.01/0.01 11:58:21
```

Response 9:

To query the standard inquiry page of the SCSI LUN backing an emulated device, enter the following:

```
q edev 111 inquiry
- Begin - EDEV 0111 - Standard Inquiry Page -
00000532 9F101002 49424D20 20202020 32313037 39303020 20202020 20202020
2E323034 37353034 31393131 34303020 20202020 20202020 00600DA0 0A000300
03200000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00323037 35003236 34303400 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00080000
- End - EDEV 0111 - Standard Inquiry Page -
```

Response 10:

To query the vital product data page 83 of the SCSI LUN backing an emulated device, enter the following:

```
q edev 111 inquiry page 83
- Begin - EDEV 0111 - Vital Product Data Page 83 -
00830024 01030010 60050763 03FFC09C 00000000 00001400 01140004 00000032
01150004 00000000
- End - EDEV 0111 - Vital Product Data Page 83 -
```

Response 11:

To query information associated with path activity on an emulated device, enter the following:

```
query edev 400 pathinfo
EDEV 0400 TYPE FBA ATTRIBUTES 2107 ALUA SUPPORTED: YES
PATHS:
FCP_DEV: 1D1D WWPN: 5005076306081588
LUN: 4064401600000000 ALUA OPTIMIZED: NO
SELECTED: 43
ADAPTER ERRORS: 1
SOFTWARE: 0
HARDWARE: 0
TRANSPORT: 1
OTHER: 0
TIMEOUT ERRORS: 0
SCSI ERRORS: 8
RESERVATION: 0
QUEUE FULL: 0
DEVICE BUSY: 4
```

## QUERY EDEVICE

```
ACA ACTIVE: 1
ABORT TASK: 0
CHECK CONDITIONS: 3
  MEDIUM: 0
  HARDWARE: 0
  NOT READY: 1
  ABORTED: 0
  OTHER: 2
LAST ERROR TIME: Apr 30 2020 08:57:00
LAST ERROR TYPE: SCSI DEVICE BUSY
LAST FAILURE TIME: Apr 30 2020 08:57:00
LAST FAILURE REASON: DIAG
```

The response for QUERY EDEVICE PATHINFO differs from QUERY EDEVICE DETAILS in that path info displays the ALUA state of the specified path. ALUA SUPPORTED can be YES, NO, or UNKNOWN, if the emulated device has yet to perform a discovery on the status.

This response also displays the number and type of errors that have occurred on the paths recently, as well as the time of the most recent error type.

If the error is an adapter error, the following types may be displayed:

### **ADAPTER SOFTWARE**

failure in the adapter microcode

### **ADAPTER HARDWARE**

failure in the adapter hardware

### **TRANSPORT DEAD**

failure in accessing the link

### **OTHER ADAPTER**

another adapter failure; for example, busy link, login failure, etc.

If the error is a SCSI command error, the type of error is listed:

- SCSI RESERVATION
- SCSI QUEUE FULL
- SCSI DEVICE BUSY
- SCSI ACA ACTIVE
- SCSI ABORT TASK
- MEDIUM CHECK COND
- HARDWARE CHECK COND
- NOT READY CHECK COND
- ABORTED CHECK COND
- OTHER CHECK COND

If the device becomes unresponsive, the error displayed is TIMEOUT ERROR.

Below the error type, the time of the most recent failure is also displayed, along with the reason for failure:

### **NONE**

No failures have occurred on this path since the last reset.

### **INIT**

The last failure occurred on this path during initialization.

### **DIAG**

The last failure occurred on this path during self-diagnostics.

### **ADAP**

The last failure occurred on this path as a result of an adapter error.

### **SCSI**

The last failure occurred on this path as a result of a SCSI command error.

**INQ**

The last failure occurred on this path as a result of a SCSI Inquiry peripheral error.

**RETRY**

The last failure occurred on this path due to too many retries of the IO operation.

Response 12:

The DETAILS response for an NVMe EDEVICE might include the following lines:

```
SEGMENT: segment number/number of segments  
HYPERPAV POOL: pool number  
PCIFUNCTION: rpfid
```

The *segment number* value is the number of the first disk block that is accessible via the emulated device. The *pool number* value is the HyperParallel Access Volume pool number that is associated with the device. The *rpfid* value is the identifier of the real PCIe function that is used to access the NVMe device.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP009E Invalid range - *range*
- HCP6704E Missing token at end of line.
- HCP8710E INQUIRY error on EDEV *rdev*
- HCP8711E EDEV *rdev* does not support VPD page *page*

## QUERY ENCRYPT



### Authorization

Privilege Class: A, C, E

### Purpose

Use the QUERY ENCRYPT command to determine the settings for host level encryption used by the z/VM hypervisor.

### Operands

#### ALL

lists settings for all the host services for which encryption is available.

#### PAGING

lists z/VM host level encryption settings for CP paging of guest memory and virtual-disk-in-storage (VDISK).

### Usage Notes

1. The QUERY ENCRYPT command lists host-level encryption configuration for both the initial system state and the current state. This information is provided for auditing purposes.
2. For a record of any user IDs that changed their initial encryption setting, refer to monitor records or to auditing records from your External Security Manager.

### Responses

Response 1:

```

Encrypt Paging settings:
  Currently: Required AES256
  At IPL: Off
Ready;
  
```

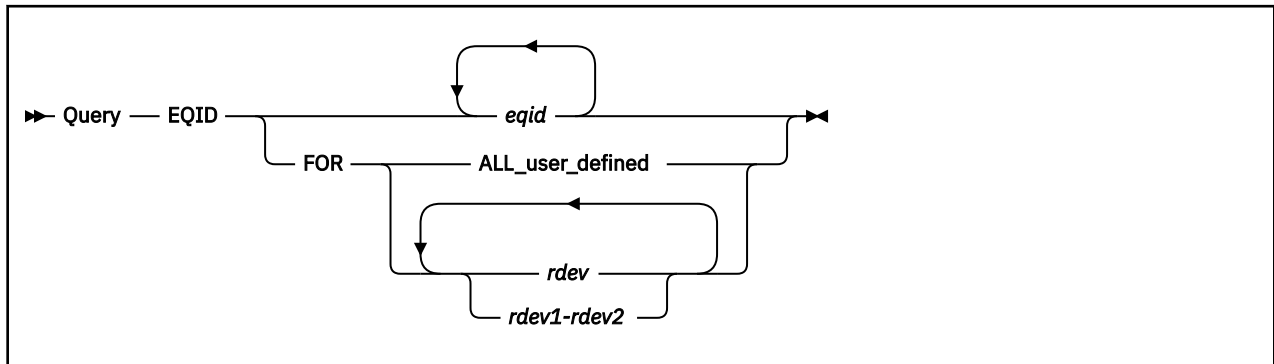
This response is issued when PAGING or ALL is specified on the QUERY ENCRYPT command. It displays the setting at the time of IPL, the present setting, and the current algorithm. The setting at the time of IPL is determined by the ENCRYPT configuration statement and can either be OFF (the default), ON, or REQUIRED. If the setting is ON or REQUIRED, the current encryption algorithm is also noted. The encryption setting can be changed with the SET ENCRYPT command; this setting can be OFF, ON, or REQUIRED. See [“SET ENCRYPT” on page 1513](#) for more information about these settings.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option|command* contains extra option(s) starting with *option*}



## QUERY EQID



### Authorization

Privilege Class: B

### Purpose

Use QUERY EQID to display the device equivalency IDs (EQIDs) for specified RDEVs, and to display the RDEVs associated with a specified equivalency ID (EQID).

### Operands

#### *eqid*

displays all RDEVs that have an EQID equal to the specified value. The *eqid* can be a string of 1–8 alphanumeric characters for a user-defined EQID or a string of 50 alphanumeric characters plus a dash ("-") for a system-generated EQID.

#### **FOR ALL\_user\_defined**

displays all RDEVs that have been assigned a user-defined EQID and the EQIDs for those RDEVs.

#### *rdev*

#### *rdev1-rdev2*

displays the EQIDs for the RDEVs within the specified range. They can be specified as a single RDEV, a range of RDEVs, or a series of both. Only RDEVs that have an EQID (either system-generated or user-defined) are displayed. RDEVs that do not exist or have no EQID are ignored. For a device in the active configuration, the device number entered can be a 4-digit hexadecimal number between X'0000' and X'FFFF'. For a device in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured. A range of devices cannot span subchannel sets.

### Usage Notes

1. The device number in the response will always be a 5-digit device number.

### Responses

Response 1:

This is a sample response to the command QUERY EQID ABCDEFGH:

```

DEVICES FOR ABCDEFGH:
00191 00192 00193 00194 00195 00196 00197 00198
00199 00205 00223 00253 00268 00271 00272 00273
00274 00288 00308 00555 00560
  
```

Response 2:

## QUERY EQID

This is a sample response to the command QUERY EQID:

```
002107900IBM7500000000000041910004000000000001000:  
DEVICES FOR 002107900IBM7500000000000041910004000000000001000:  
04392
```

Response 3:

This is a sample response to the command QUERY EQID FOR ALL:

```
00001: EQID = POOL1  
00100: EQID = POOL2  
00990: EQID = POOL3  
00991: EQID = POOL3  
00992: EQID = POOL3  
0BB01: EQID = OSADEV1  
0FFF3: EQID = OSADEV2
```

Response 4:

This is a sample response to the command QUERY EQID FOR 190:

```
00190: EQID = MYOSA
```

Response 5:

This is a sample response to the command QUERY EQID FOR 190-195:

```
00190: EQID = MYOSA  
00193: EQID = 002107900IBM7500000000000041910004000000000000FFFF  
00195: EQID = POOL1
```

Response 6:

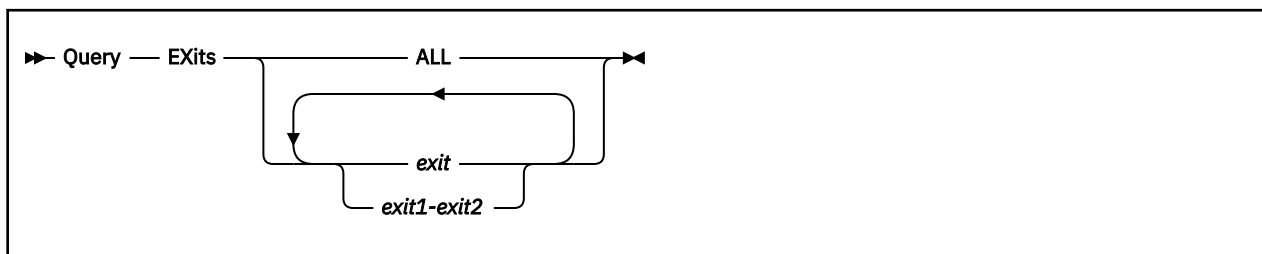
This is a sample response to the command QUERY EQID  
00020E662C220E6-0000-0201-0005F8C500000000013FFFFF:

```
11:58:46 Devices for 00020E662C220E6-0000-0201-0005F8C500000000013FFFFF:  
11:58:46 07F00  
Ready; T=0.01/0.01 11:58:46
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP040E Device *rdev* does not exist
- HCP048E No device with EQID *eqid* exists.
- HCP048E No EQID exists for *rdev*.
- HCP1003E *rdev* was not found.

## QUERY EXITS



### Authorization

Privilege Class: A, C, E

### Purpose

Use QUERY EXITS to display status and usage statistics information for one or more exit points.

### Operands

#### ALL

tells CP to display information about all existing exit points.

#### exit

#### exit1-exit2

is the number of the exit point for which you want information. Each *exit* must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single exit number, a range of exit numbers, or any combination thereof.

### Usage Notes

- For a static exit, CP does not create CP exit control blocks until you associate one or more entry points or external symbols with that exit point. If you try to issue a QUERY EXITS command against such an exit point, CP issues message HCP2752E as the response to your QUERY EXITS command. For a dynamic exit, QUERY EXITS responds with the definition of the exit, even if there are no entry points associated with it.
- To create a new dynamic exit point, use the DEFINE EXIT command (see [“DEFINE EXIT”](#) on page 176).
- To activate an exit point:
  - While defining it, use the ENABLE operand of the ASSOCIATE EXIT command (see [“ASSOCIATE EXIT”](#) on page 20).
  - After defining it, use the ENABLE EXITS command (see [“ENABLE EXITS”](#) on page 465).
- To display the address of the CP exit block for a specific exit point, use the LOCATE XITBK command (see [“LOCATE XITBK”](#) on page 649).
- To display the address of the CP indirect call locator block for a specific exit point, use the LOCATE ICLBK command (see [“LOCATE ICLBK”](#) on page 614).
- To deactivate an exit point:
  - While defining it, use the DISABLE operand of the ASSOCIATE EXIT command (see [“ASSOCIATE EXIT”](#) on page 20).
  - After defining it, use the DISABLE EXITS command (see [“DISABLE EXITS”](#) on page 319).
- To change the definition of an existing dynamic exit point, or to remove the exit point from the system, use the MODIFY EXIT command (see [“MODIFY EXIT”](#) on page 700).

8. For more information about user-defined exit points, see [Dynamic Exit Points](#) in *z/VM: CP Exit Customization*.

## Responses

Response 1:

To display the current information for exit point X'77', enter the following:

```
query exits 77
Exit Status      Calls   Returns Seconds
0077 Enabled       3       3 0.000633
      EPNAME      Attempts Calls   Seconds
      HCPSRC00    3       3 0.000598
Ready;
```

In this example,

*Lines 1 and 2 – Information about the Exit Point*

### Exit

shows the number of the exit point.

### Status

tells you whether CP is accepting requests to process this exit point. If "Enabled", CP is accepting requests; if "Disabled", CP is not accepting requests.

### Calls

tells you how many times CP has entered this exit point.

### Returns

tells you how many times this exit point has returned control to CP.

**Note:** If the number of times entered equals the number of times exited (as in this example), then you know the exit point is not currently active. If the number of times entered is greater than the number of times exited, then you know the exit point is currently processing a request.

### Seconds

shows the total elapsed time CP used to process this exit point. This number includes the time it took for CP to set up the exit point plus the total elapsed times that CP used to process all calls to all entry points.

*Lines 3 and 4 – Information about Each Entry Point Associated With the Exit Point*

### EPNAME

shows you each entry point associated with this exit point, in the order that CP calls them. (The exit point in this example only has 1 entry point, HCPSRC00.)

### Attempts

shows you how many times CP has tried to call this entry point.

### Calls

shows you how many times CP succeeded in calling this entry point.

**Note:** If the number of attempts equals the number of calls, you know that CP had no trouble finding the entry point and passing control over to it. If the number of attempts is greater than the number of calls, then CP could not locate the entry point during 1 or more attempts to call it.

### Seconds

shows the total elapsed time CP used to process this specific entry point. (In this example, it is the total time CP used to process all 3 calls to entry point HCPSRC00.)

Response 2:

To display the current information for CP exit point X'D5', enter the following:

```
query exits d5
Exit Status      Calls   Returns Seconds
```

```
00D5 Enabled      8      8 10.008833
      EPNAME      Attempts  Calls  Seconds
      HCPSRC00     8      8 0.002598
      HCPSRCZZ     8      0 0.000000
      HCPSRC08     8      4 0.002833
      HCPSRC99     6      6 10.002668
Ready;
```

In this example, the first 2 lines show that CP:

- Is accepting requests to process this exit point.
- Entered and exited this exit point 8 times, which means the exit point is not currently active.
- Used a little more than 10 seconds to set up this exit point and process all calls to all its entry points.

The remaining lines show that:

- Exit point X'D5' currently has 4 entry points (HCPSRC00, HCPSRCZZ, HCPSRC08, and HCPSRC99).
- CP tried to call HCPSRC00 8 times and succeeded in each case.
- CP tried to call HCPSRCZZ 8 times, but none of the attempts were successful. (This probably means that someone did not load HCPSRCZZ yet.)
- CP tried to call HCPSRC08 8 times. Only 4 of those attempts were successful. (This can happen when someone loads an entry point after associating it with an exit point.)
- CP tried to call HCPSRC99 6 times and succeeded in each case. However, there are 2 odd things about HCPSRC99:
  1. CP called the other entry points 8 times, but only called HCPSRC99 6 times. (This probably means that someone associated the entry point with the exit point sometime after the other 3 entry points.)
  2. The elapsed time to process HCPSRC99 is substantially longer than that of the other 3 entry points. (Unless this entry point is substantially larger than the others, this probably means someone should investigate this entry point.)

Response 3:

To display the current information for CP exit point X'1234', enter the following:

```
query exits 1234
Exit Status      Calls  Returns Seconds
1234 Enabled      3      3 0.000803
      Location      Instruction Resolution
      HCPXMGMS+0004 41700000  RESOLVE
      Defined by: OPERATOR at 13:27:54 on 09/03/98
      PARM R9
      RC
      EPNAME      Attempts  Calls  Seconds
      EXITSTDP    3      3 0.000785
Ready;
```

In this example, the first 2 lines show that CP:

- Is accepting requests to process this exit point.
- Entered and exited this exit point 3 times, which means the exit point is not currently active.
- Used a little less than 1 millisecond to set up this exit point and process all calls to its associated entry point.

The next five lines show that:

- The exit point was defined dynamically.
- The exit point is located at 4 bytes past entry point HCPXMGMS, where instruction X'41700000' occurs.
- The resolution of the entry point associated with the exit was performed when the association was made between them.
- The exit point was defined by user OPERATOR on September 3, 1998, at 13:27:54 local time.
- The exit receives two parameters, namely the contents of general registers 9 and 12 at the point of exit, in addition to the three parameters that are passed to every dynamic exit.

## QUERY EXITS

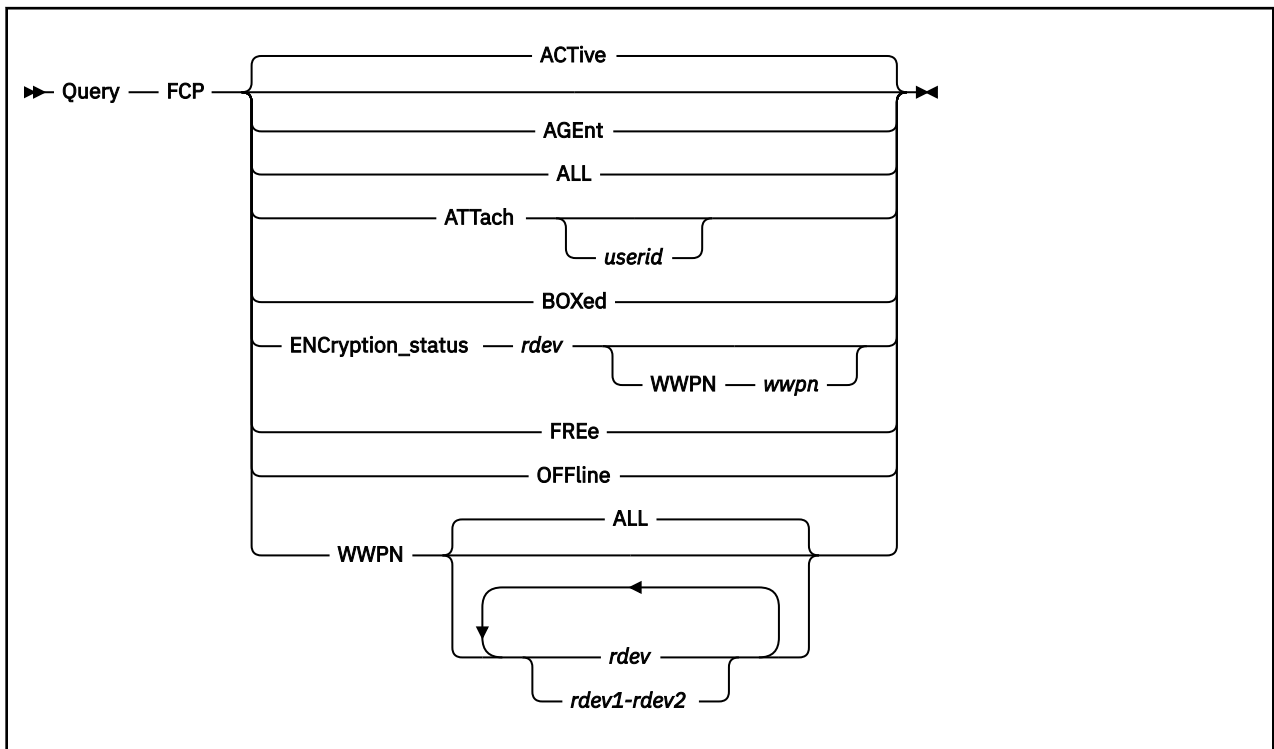
The last two lines show that:

- Exit point X'1234' currently has 1 entry point (EXITSTDP).
- CP tried to call EXITSTDP 3 times and succeeded in each case.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP2752E Exit number *exit* does not exist
- HCP2752E Exit numbers do not exist
- HCP2752E Exit numbers *exit1-exit2* do not exist
- HCP6704E Missing token at end of line
- HCP6706E Invalid CP Exit number - *exit*

## QUERY FCP



### Authorization

Privilege Class: B

### Purpose

Use QUERY FCP to display the status of real FCP Adapter devices.

### Operands

#### ACTIVE

indicates active FCP Adapter devices. An active device is one that is in use by a user or the system and is specified as neither FREE nor OFFLINE.

#### AGEnt

indicates the Fibre Channel Protocol Agent device. The FCP Agent device is used to control FCP configuration.

#### ALL

indicates all FCP Adapter devices attached to the system. The status is displayed in the order of ACTIVE, FREE, OFFLINE, and AGENT. You receive the same response as you would receive if you entered all of the following commands:

- query fcp active
- query fcp free
- query fcp offline
- query fcp agent

**ATTach *userid***

indicates the FCP devices that are dedicated to a user on the system. If you specify *userid*, only FCP devices attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all FCP devices dedicated to any user on the system. Attached devices are always active.

**BOXed**

indicates the FCP devices that are unavailable (usually because of a hardware condition). A boxed adapter can also be active, free, or offline.

**ENCryption\_status**

indicates the encryption status of a target WWPN (if a WWPN is specified) or the encryption status of all of the target WWPNs that are associated with the specified FCP device.

**WWPN *wwpn***

indicates a target WWPN on the storage controller (associated with the HBA).

**FREE**

indicates all free FCP devices. A free FCP device is one that is not currently in use by a user or the system and is specified as neither ACTIVE nor OFFLINE.

**OFFline**

indicates all offline FCP devices. An offline FCP device is one that is specified as neither ACTIVE nor FREE.

**WWPN**

indicates for each FCP device the NPIV worldwide port name (WWPN) associated with the device and the physical WWPN associated with the device's channel path. The status of the device will also be displayed (ATTACHED, FCP AGENT, FREE, OFFLINE, BOXED).

**Usage Notes**

1. There is a Linux Access Control Table that is used to control FCP LUN access. VM userids are entered in this table with corresponding LUNs. For details, refer to Linux documentation.
2. A VIRTUAL version of this command, QUERY VIRTUAL FCP, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL FCP” on page 1297](#).

**Responses**

If the system is using an FCP device that is in an intervention-required state and you enter QUERY FCP, you see a response like this:

```
FCP 0300 has intervention required.
```

0300 is the real device number of the FCP device that needs the operator's intervention.

**Note:** This response will be preceded by one of the following.

Response 1:

If you enter QUERY FCP, QUERY FCP ACTIVE, or QUERY FCP ATTACH and there is an FCP device attached to the system, you see something like this:

```
q fcp active
FCP 0480 ATTACHED TO LINUX1 0480 CHPID 80
FCP 0481 ATTACHED TO LINUX2 0481 CHPID 80 USERACCESSID = userid
FCP 0482 ATTACHED TO LINUX2 0482 CHPID 80
FCP 0700 ATTACHED TO TEST1 0500 CHPID F1
FCP 0701 ATTACHED TO TEST2 0500 CHPID F1
Ready;
```

The first number is the real device number of the FCP device. The operand might also read BOX/ATTC, which means that although the device is attached and dedicated to a user, it is unavailable. LINUX1, LINUX2, TEST1, and TEST2 are user IDs to which the FCP devices are dedicated. The number following



the user IDs is the virtual device number which each virtual machine uses to refer to an FCP device. The number following the CHPID is the real CHPID (channel path ID) associated with the device.

If the FCP subchannel being displayed was attached using the Attach USERACCESSID option, the USERACCESSID will be displayed as part of the response.

Response 2:

If you enter QUERY FCP, QUERY FCP ACTIVE, or QUERY FCP ATTACH as above, and NPIV is enabled on the FCP subchannels being queried, you see something like this:

```
q fcp active
FCP 2FC1 ATTACHED TO NP0RT      2FC1 CHPID 23
   WWPN C05076FFFD8018EC
FCP 2FC2 ATTACHED TO NP0RT      2FC2 CHPID 23
   WWPN C05076FFFD8018F0
FCP 2FC3 ATTACHED TO NP0RT      2FC3 CHPID 23
   WWPN C05076FFFD8018F4
Ready;
```

The number following WWPN is the unique world wide port name associated with the FCP device.

Response 3:

If you enter a QUERY FCP FREE command and there is a free FCP device attached to the system, you see something like this:

```
q fcp free
FCP 0487 FREE      , FCP 04A1 FREE      , FCP 04A2 FREE      , FCP 04A3 FREE
FCP 04A4 FREE      , FCP 04A5 FREE      , FCP 04A9 FREE      , FCP 04AA FREE
FCP 04AB FREE      , FCP 04BC FREE      , FCP 04BD FREE      , FCP 04BE FREE
FCP 04BF FREE
Ready;
```

This response shows the real device number for each free FCP device.

The response for a QUERY FCP OFFLINE or QUERY FCP BOXED command is in the same format as the response for a QUERY FCP FREE command.

Response 4:

When you enter a QUERY FCP ALL command, CP displays the same responses as it would display if you had entered the QUERY FCP ACTIVE, QUERY FCP FREE, and QUERY FCP OFFLINE commands.

Response 5:

If CP cannot find an FCP device of the type you specify, you see a message like this:

```
q fcp boxed
A BOXED FCP was not found.
Ready;
```

In this response, BOXED may be replaced with active, attached, offline, or free, depending on what you specified on the command.

Response 6:

When you enter a QUERY FCP AGENT command and there is an FCP AGENT attached to the system, you see something like this:

```
q fcp agent
FCP 0280 is an FCP Agent.
Ready;
```

or, in the case where there is no agent, you will see:

```
q fcp agent
An FCP Agent was not found.
Ready;
```

Response 7:

## QUERY FCP

When you enter a QUERY FCP WWPN command, you see something like this:

```
q fcp wwpn 2f00-2fff
FCP 2F00      NPIV WWPN NONE
      CHPID 24 PERM WWPN C05076FFFD8018FD
      OFFLINE

FCP 2F08      NPIV WWPN NONE
      CHPID 25 PERM WWPN C05076FFFD8018FE
      FREE

FCP 2FF0      NPIV WWPN C05076FFFD8018EC
      CHPID 26 PERM WWPN C05076FFFD8018FF
      ATTACHED TO NPORT
Ready;
```

Response 8:

When you enter a QUERY FCP WWPN command and there are no FCP devices that exist related to the query, you see something like this:

```
q fcp wwpn 1f00-1fff
A valid FCP device was not found.
Ready;
```

Response 9:

When you enter a QUERY FCP ENCRYPTION\_STATUS command for an FCP device, you see something like this:

```
q fcp enc 8181
FCP 8181 WWPN C05076FFFD8018F0 AVAILABLE
FCP 8181 WWPN C05076FFFD8018F2 AUTHENTICATED
FCP 8181 WWPN C05076FFFD8018F4 ENCRYPTED
Ready;
```

Response 10:

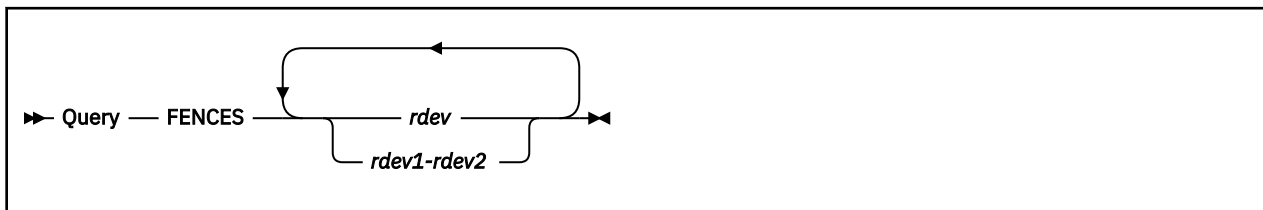
When you enter a QUERY FCP ENCRYPTION\_STATUS command for an FCP device and a target WWPN, you see something like this:

```
q fcp enc 8181 C05076FFFD8018F4
FCP 8181 WWPN C05076FFFD8018F4 ENCRYPTED
Ready;
```

## Messages

- HCP003E Invalid option - *option*
- HCP006E Invalid device type - *rdev*
- HCP007E Invalid userid - *userid*
- HCP263E Too many operands were supplied for this command.
- HCP6004E This command is not supported on your system.
- HCP6706E Invalid address range - *range*

## QUERY FENCES



### Authorization

Privilege Class: B

### Purpose

Use QUERY FENCES to display the following:

- A list of internal fences created by an IBM DASD subsystem, as a result of internal errors, to inhibit use of various subsystem resources.
- Devices which have been put into the soft fence state to protect the devices from read or write operations.

### Operands

*rdev*

*rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers of the devices being queried.

### Usage Notes

1. The QUERY FENCES command is only valid for IBM DASD subsystems. An error message is generated if the specified device is not attached to an IBM DASD subsystem.

### Responses

Response 1:

```
rdev No fences exist for this device.
```

This response is issued when no fences exist for the specified subsystem, and is repeated for each specified device.

*rdev*

is the real device number of a device being queried.

Response 2:

```
rdev type fenced:
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
or
rdev type fenced:
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

This response is issued when fences do exist for the specified subsystem, and is repeated for each specified device.

## QUERY FENCES

### ***rdev***

is the real device number of a device being queried.

### ***type***

Storage path | Channel attachment | Device attachment

### **XXXXXXXX**

Four hexadecimal bytes of storage path status data

For 3990 Model 3, 16 bytes of storage path status data is returned. For all other models of IBM DASD subsystems, 24 bytes of storage path status is returned.

Response 3:

```
rdev is in a Soft Fenced State.
```

This response is issued when the soft fence state exists for the device and is repeated for each specified device in that state.

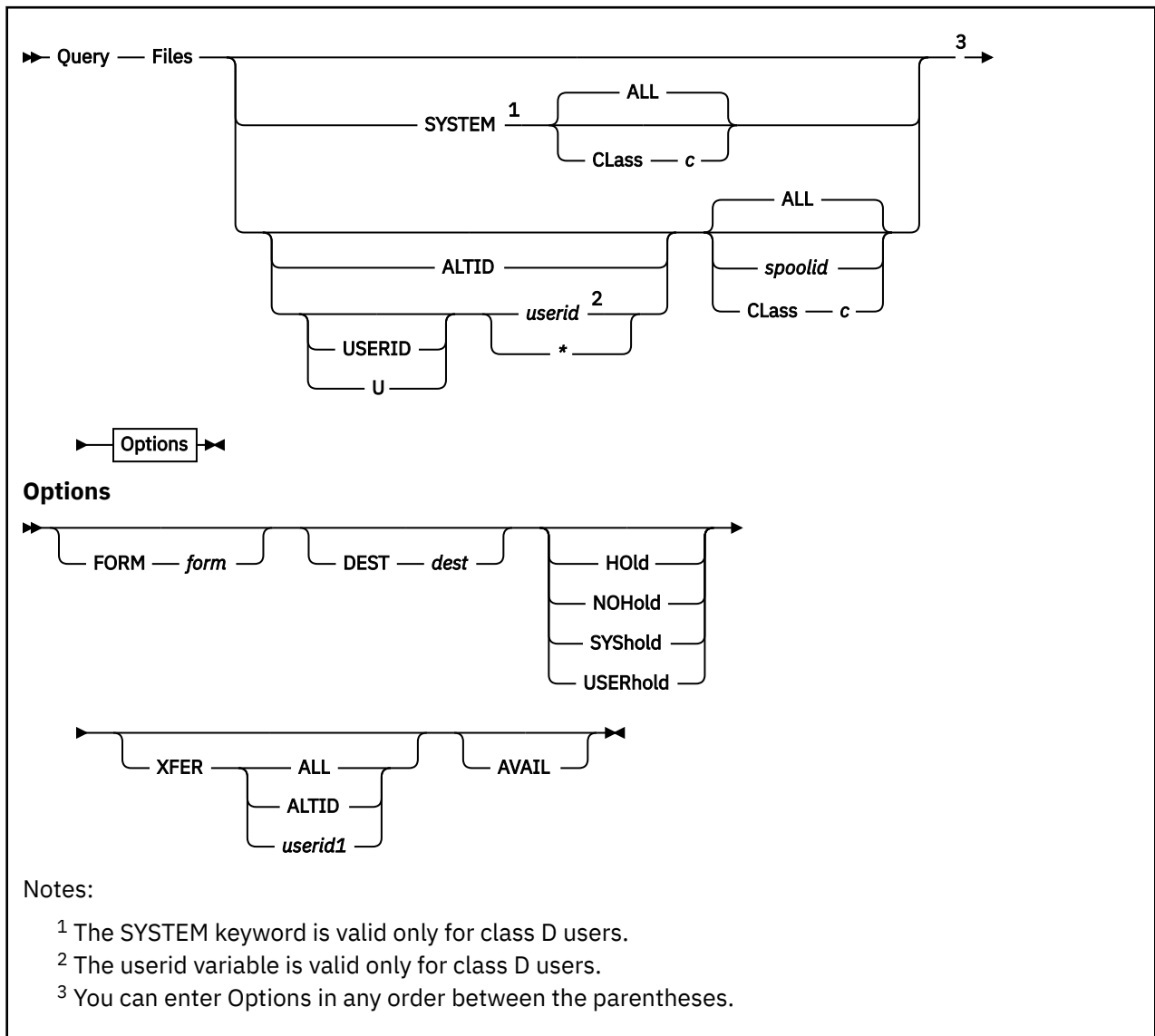
### ***rdev***

is the real device number of a device being queried.

## Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*
- HCP1109E The *command* command cannot be processed for device *rdev* because the device is the secondary in a duplex pair.

# QUERY FILES



## Authorization

Privilege Class: D, G

## Purpose

Use QUERY FILES to display a count of spool files. The count includes the following:

- Files open on virtual devices
- Files being processed on real devices.

## Operands

### SYSTEM

obtains a count of all spool files in the system. The SYSTEM option is the default for class D users, and can only be used by class D users. It is not valid with the *spoolid* option.

## QUERY FILES

### USERID

#### U

indicates that the next parameter is a userid. Class G users can specify only an asterisk (\*) following USERID or U. U is the only abbreviation allowed for USERID.

### ALTID

#### *userid*

#### \*

identifies the user whose spool files are to be queried. You can specify *userid* as an asterisk (\*) for information on your own files. You can specify ALTID for information on files belonging to your alternate user ID. For a class G user, the asterisk (\*) is the default.

### ALL

queries all of your files and is the default.

### Class c

designates an existing spool file class; the operand *c* is a 1-character alphanumeric field from A through Z or 0 through 9.

### *spoolid*

identifies the file to be queried; this number is assigned by the system. This option is not valid with the SYSTEM keyword, nor can it be used if a class D user is defaulting to SYSTEM. This option can only be used by a class D user if a user ID or an asterisk (\*) is specified in the command.

### FORM *form*

indicates that you want to count the number of spool files with the specified form. The operand *form* is one to eight characters long. For the class G user, *form* is the user form name. For the class D operator, *form* is the user form name when asterisk (\*) is specified. Otherwise, *form* is the operator form number.

### DEST *dest*

indicates that you want to count the number of spool files with the specified destination value. The operand *dest* is a one to eight character name.

### HOLD

#### NOHOLD

#### SYSHOLD

#### USERHOLD

indicates that you want to query files that are in a particular hold status. Specify HOLD to query files that are in user or system hold. Specify USERHOLD to query files that have only user hold in effect. Specify SYSHOLD to query files that have only system hold in effect. Specify NOHOLD to query files that are not in user or system hold.

A file that is in hold status cannot be processed by a spooling device; you can use CHANGE to alter the file's hold status.

### XFER ALL

### XFER ALTID

### XFER *userid1*

indicates that you want to query files generated by the specified user ID or ALTID and owned by another user ID. Class D users may use the SYSTEM option to obtain a count of all files in the system owned by a user ID other than the originating user ID. XFER *userid1* displays a count of the spool files generated by the specified user ID or ALTID and which are owned by the user specified as *userid1*. Files generated and owned by the same user are not counted. XFER ALTID displays a count of files generated by the specified user ID and owned by ALTID. XFER ALL displays a count of all spool files generated by the specified user ID or ALTID and which are not owned by the originating user ID.

### AVAIL

displays only a count of available files, not those files that are open or in use.

## Usage Notes

1. Privilege class G users can enter QUERY FILES to display the number of spool files they own or have generated. Privilege class D users can enter QUERY FILES to display all spool files in the system, or all files that belong to a particular user of that system.
2. If the user ID option is used with the XFER option by a class D user, the specified user ID is treated as the originator of the files. The result is a count of the files generated by the first user, and includes those files owned by the user specified after the XFER option.

For example, if a class D user enters `query files user1`, a count is returned of the files that USER1 owns.

If a class D user enters `query files xfer user2`, a count is returned of the files that USER2 owns, but did not generate.

If a class D user enters `query files user1 xfer user2`, a count is returned of the files generated by USER1 and owned by USER2.

3. In an SSI cluster, if the AVAIL operand is not specified, this command displays information about all files in the system on which the command is processed. If the AVAIL operand is specified, this command does not display information about files that are locked by another system.

## Responses

Response 1:

For the class D user:

```
FILES: {nnnnnnn} RDR, {nnnnnnn} PRT, {nnnnnnn} PUN
       {      NO}   {      NO}   {      NO}
```

where *nnnnnnn* is the number of spool files of each type.

For the class G user:

```
FILES: {nnnn} RDR, {nnnn} PRT, {nnnn} PUN
       { NO}     { NO}     { NO}
```

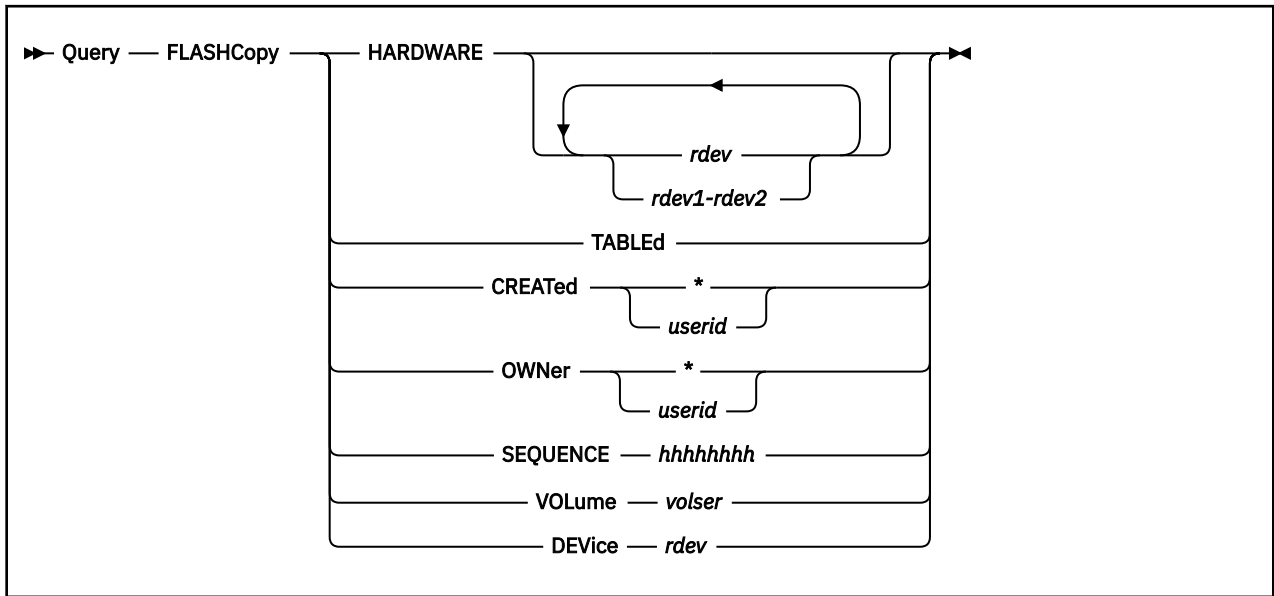
where *nnnn* is the number of spool files of each type.

**Note:** If a class G user issues QUERY FILES (or if a privileged user issues QUERY FILES \*), and if that user has more than 9999 rdr, prt, or pun files, the response will still only show 9999 files.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP053E *userid* not in CP directory
- HCP364E No alternate *userid* is currently set.
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP849E A user ID must be specified with the spoolid
- HCP1558E Command processing cannot complete.

## QUERY FLASHCOPY



### Authorization

Privilege Class: B

### Purpose

This command allows interrogation of persistent FlashCopy relationships in the hardware and the Persistent FlashCopy Relationship Table. Each FlashCopy relationship is identified with a unique sequence number that is derived from the TOD clock when the relationship is established.

The Persistent FlashCopy Relationship Table is maintained by z/VM to preserve additional information regarding FlashCopy relationships. Note that this table is NOT maintained over system IPLs and is intended to be used as supplemental information regarding FlashCopy relationships.

### Operands

#### **HARDWARE**

specifies the real hardware devices that are to be interrogated. The resulting display will indicate all the persistent relationships that exist on the devices.

#### **TABLEd**

displays all additional information regarding persistent FlashCopy relationships that were established during the current system IPL session.

#### **CREATed \***

##### **CREATed *userid***

displays additional information regarding persistent FlashCopy relationships that were established by the specified user during the current system IPL session.

#### **OWNer \***

##### **OWNer *userid***

displays additional information regarding persistent FlashCopy relationships that were established on minidisks owned by the specified user during the current system IPL session.

#### **SEQUENCE *hhhhhhh***

displays additional information regarding persistent FlashCopy relationships that were established and assigned the specified sequence number during the current system IPL session.



**VOLume volser**

displays additional information regarding persistent FlashCopy relationships that were established on the specified real volume during the current system IPL session.

**DEVICE rdev**

displays additional information regarding persistent FlashCopy relationships that were established on the specified real device during the current system IPL session.

**Usage Notes**

1. The contents of the Persistent FlashCopy Relationship Table are lost when the system is IPLed.
2. Entries might exist in the list for relationships that have been partially withdrawn by this or another LPAR.
3. Entries will not exist for relationships that have been completely withdrawn.
4. When the number of tracks remaining to be background copied is nonzero and you want to issue the FLASHCOPY WITHDRAW command, you must use the FORCE option.
5. If flag X'0002' is set to one, the number of remaining tracks to be copied is reported as '?'.  
 6. A VIRTUAL version of this command, QUERY VIRTUAL FLASHCOPY, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see ["QUERY VIRTUAL FLASHCOPY"](#) on page 1300.

**Responses**

Response 1: QUERY FLASHCOPY HARDWARE

```

q flashcopy hardware
HCP2466I Device 5100 - No FlashCopy relationships reported.
HCP2466I Device 5101 - No FlashCopy relationships reported.
HCP2466I Device 5C80 - No FlashCopy relationships reported.
HCP2466I Device 5C81 - No FlashCopy relationships reported.
HCP2466I Device FCE0 - No FlashCopy relationships reported.
HCP2466I Device FCE1 - No FlashCopy relationships reported.
HCP2466I Device FEE0 - No FlashCopy relationships reported.
HCP2466I Device FEE1 - No FlashCopy relationships reported.
Ready(02466); T=0.01/0.01 15:18:35

q flashcopy hardware 5100-5103
HCP2466I Device 5100 - No FlashCopy relationships reported.
HCP2466I Device 5101 - No FlashCopy relationships reported.
HCP040E Device 5102 does not exist
HCP040E Device 5103 does not exist
Ready(00040); T=0.01/0.01 15:19:24

flashcopy establish source 200 target 300
Command complete: FLASHCOPY 0200 0 99 TO 0300 0 99
FLASHCOPY RELATIONSHIP 4B145C34 ESTABLISHED
Ready; T=0.01/0.01 15:19:55

flashcopy establish source 210 target 310
Command complete: FLASHCOPY 0210 0 99 TO 0310 0 99
FLASHCOPY RELATIONSHIP 4B14700A ESTABLISHED
Ready; T=0.01/0.01 15:20:00

q flashcopy hardware
-----SOURCE-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
4B145C34 0800 5100 PACK01 100/00 5101 PACK02 100/00 0/1500
4B14700A 0800 5100 PACK01 200/00 5101 PACK02 200/00 0/1500
4B145C34 8800 5100 PACK01 100/00 5101 PACK02 100/00 0/1500
4B14700A 8800 5100 PACK01 200/00 5101 PACK02 200/00 0/1500
HCP2466I Device 5C80 - No FlashCopy relationships reported.
HCP2466I Device 5C81 - No FlashCopy relationships reported.
HCP2466I Device FCE0 - No FlashCopy relationships reported.
HCP2466I Device FCE1 - No FlashCopy relationships reported.
HCP2466I Device FEE0 - No FlashCopy relationships reported.
HCP2466I Device FEE1 - No FlashCopy relationships reported.
Ready(02466); T=0.01/0.03 15:20:06
    
```

## QUERY FLASHCOPY

**Note:** If an RDEV in a FlashCopy relationship is not attached to this z/VM image, you will receive a display such as:

```
q flashcopy hardware
-----SOURCE-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
62320413 E818 ssid *lsua*          0/00 FCE2 GESE01          0/00 50084/50085
```

When *\*lsua\** is displayed for the VOLSER, the RDEV value that is displayed is the partner's subsystem ID and the *\*lsua\** value is the partner's logical subsystem number and unit address, respectively.

Example:

```
q flashcopy hardware
-----SOURCE-----
SEQUENCE FLGS RDEV VOLSER CC...CC/HH RDEV VOLSER CC...CC/HH REMAINING/TOTAL
62320413 E818 0202 *0200*          0/00 FCE2 GESE01          0/00 50084/50085
```

The FLGS field of the QUERY FLASHCOPY HARDWARE response is in hexadecimal and has the following meaning:

Flag Bit	Definition
8000	0 = Source, 1 = Target
4000	1 = Full Volume Relationship
2000	1 = Background copy pending (NOCOPY)
1000	1 = Background copy in progress (BACKGNDCOPY)
0800	1 = Persistent Relationship (ESTABLISH)
0400	1 = Target extents are write protected (NOTGTWRITE)
0200	1 = Validation required
0100	1 = Change recording is active (CHGRECORD)
0080	1 = Reversible relationship (not set by z/VM)
0040	1 = Source write inhibited (not set by z/VM)
0020	1 = Virtual flash initialize relationship
0010	1 = Space efficient target volume or extent
0008	1 = Fail if no space (FAILNOSPACE)
0004	1 = Relationship is failed
0002	1 = Number of remaining tracks to copy is unknown
0001	1 = Source volume write inhibited due to full repository

For the --SOURCE-- and --TARGET-- areas, the CC...CC/HH fields are the decimal cylinder and head numbers for the beginning of the respective extent. (Note that the CC...CC cylinder field could contain as many as nine characters.) The REMAINING field contains the number of unmodified source tracks that have yet to be copied to the target, and the TOTAL field contains the total number of tracks in the extent.

Response 2: QUERY FLASHCOPY TABLED

```
q flashcopy tabled
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
4B145C34 2008-04-30 15:19:55 5100> PACK01 RWS RWS 0200
4B145C34 2008-04-30 15:19:55 >5101 PACK02 RWS RWS 0300
4B14700A 2008-04-30 15:20:00 5100> PACK01 RWS RWS 0210
```

```
4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS RWS 0310
Ready; T=0.01/0.01 15:20:40
```

The RDEV> indicates a source relationship and the >RDEV indicates a target relationship.

The CREATOR is the VM user ID that issued the FLASHCOPY ESTABLISH command.

The OWNER and VDEV are the minidisk owner's information. For example, if you did this:

```
LINK MAINT 190 999 RR
FLASHCOPY ESTABLISH SOURCE 999 TARGET 200
QUERY FLASHCOPY CREATED RWS
```

you would get back

```
q flashcopy vol rws541
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
C1D5410F 2008-01-21 18:23:40 5C81> PACK03 RWS MAINT 0190
```

rather than

```
q flashcopy vol rws541
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
C1D5410F 2008-01-21 18:23:40 5C81> PACK03 RWS RWS 0999
```

### Response 3: QUERY FLASHCOPY CREATED

```
q flashcopy created rws
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
4B145C34 2008-04-30 15:19:55 5100> PACK01 RWS RWS 0200
4B145C34 2008-04-30 15:19:55 >5101 PACK02 RWS RWS 0300
4B14700A 2008-04-30 15:20:00 5100> PACK01 RWS RWS 0210
4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS RWS 0310
Ready; T=0.01/0.01 15:20:53
```

### Response 4: QUERY FLASHCOPY OWNER

```
q flashcopy owner rws
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
4B145C34 2008-04-30 15:19:55 5100> PACK01 RWS RWS 0200
4B145C34 2008-04-30 15:19:55 >5101 PACK02 RWS RWS 0300
4B14700A 2008-04-30 15:20:00 5100> PACK01 RWS RWS 0210
4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS RWS 0310
Ready; T=0.01/0.01 15:21:33
```

### Response 5: QUERY FLASHCOPY VOLUME

```
q flashcopy volume pack02
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
4B145C34 2008-04-30 15:19:55 >5101 PACK02 RWS RWS 0300
4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS RWS 0310
Ready; T=0.01/0.01 15:21:45
```

### Response 6: QUERY FLASHCOPY SEQUENCE

```
q flashcopy sequence 4B14700A
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
4B14700A 2008-04-30 15:20:00 5100> PACK01 RWS RWS 0210
4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS RWS 0310
Ready; T=0.01/0.01 15:22:13
```

### Response 7: QUERY FLASHCOPY DEVICE

```
q flashcopy device 5101
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
4B145C34 2008-04-30 15:19:55 >5101 PACK02 RWS RWS 0300
4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS RWS 0310
Ready; T=0.01/0.01 15:22:41
```

### Response 8: If Information Is Not Available

## QUERY FLASHCOPY

In some cases, information might no longer be available, and you could see \*UNKNOWN.

```
q flashcopy created rws
SEQUENCE ---DATE--- --TIME--  RDEV  VOLSER  CREATOR  OWNER  VDEV
4B145C34 2008-04-30 15:19:55 5100> PACK01 RWS      *UNKNOWN 0200
4B145C34 2008-04-30 15:19:55 >5101 PACK02 RWS      RWS      0300
4B14700A 2008-04-30 15:20:00 5100> PACK01 *UNKNOWN RWS      0210
4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS      RWS      0310
Ready; T=0.01/0.01 15:20:53
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP040E Device *rdev* does not exist
- HCP2465I No matches found.
- HCP2466I Device *rdev* - No FlashCopy relationships reported.

## QUERY FRAMES

► Query — FFrames ◄

### Authorization

Privilege Class: A, B, E

### Purpose

Use QUERY FRAMES to display the status of host real storage.

### Responses

```
All frames:
Configured=n Real=n Usable=n Offline=n
Pageable=n NotInitialized=n GlobalClearedAvail=n
LocalClearedAvail=n LocalUnclearedAvail=n
LockedPageable=n% LockedWarning=--% LockedFail=--%
IOATSize=n IOATUsed=n% IOATMaxUsed=n% IOATWarning=--%
Permanent=n Reconfigurable=n Vacating=n
```

```
Frames < 2G:
GlobalUnclearedAvail=n Pageable=n LogicalFreeStorage=n
RealFreeStorage=n LockedRS=n LockedCmd=n
MinidiskCache=n Nucleus/Prefix=n Trace=n Other=n
```

```
Frames > 2G:
GlobalUnclearedAvail=n Pageable=n LogicalFreeStorage=n
RealFreeStorage=n LockedRS=n LockedCmd=n
MinidiskCache=n Other=n
```

### All Frames

#### Configured

indicates the amount of real storage, in frames, that is assigned to the logical partition in which z/VM is running. The amount of storage represented by this operand is equal to the CONFIGURED= value in the response to the QUERY STORAGE command.

#### Real

indicates the amount of real storage, in frames, that is assigned to the logical partition in which z/VM is running. The Real= and Configured= values are always equal.

#### Usable

indicates the number of frames that are online to z/VM.

Usable = Permanent + Reconfigurable - Offline.

Usable = Configured - Offline.

**Note:** If the IPL parameter STORE=n was used when IPLing a second-level z/VM system in a configuration without standby or reserved storage, then Usable <= Configured - Offline.

The amount of storage represented by Usable + Offline equals the STORAGE=n value in the response to the QUERY STORAGE command.

#### Offline

indicates the number of frames taken offline because of error.

#### Pageable

indicates the number of frames available for paging.

**NotInitialized**

indicates the number of frames that have not been initialized. Because all storage below 2 GB is initialized, this refers only to storage above 2 GB.

**GlobalClearedAvail**

indicates the number of frames on the Global Cleared Avail list, which identifies frames that have been cleared to 0 and are available for use by CP. These frames may be above or below 2 GB. There is only one Global Cleared Avail list, regardless of the number of processors.

**LocalClearedAvail**

indicates the sum of the number of frames on all the Local Cleared Avail lists. There is one Local Cleared Avail list for each processor. It identifies frames that have been cleared to 0 and are available for use by CP. The frames may be above or below 2 GB.

**LocalUnclearedAvail**

indicates the sum of the number of frames on all the Local Uncleared Avail lists. There is one Local Uncleared Avail list for each processor. It identifies frames that have not been cleared to 0 and are available for use by CP. The frames may be above or below 2 GB.

**LockedPageable**

is the percentage of pageable storage currently pinned.

**LockedWarning**

is specified by the system programmer in the system configuration file. Message HCP401I is sent when the percentage of locked pageable storage exceeds this percent.

The percent value for LockedWarning is displayed as -- if STORAGE LOCKING is not specified in the system configuration file.

**LockedFail**

is specified by the system programmer in the system configuration file. PCIe function translation fails when this limit is reached. Message HCP401I is sent when this happens.

The percent value for LockedFail is displayed as -- if STORAGE LOCKING is not specified in the system configuration file.

**IOATSize**

is the number of frames reserved by the system program in the system configuration file for the IOAT subpool.

**IOATUsed**

is the percentage of the IOAT subpool currently used.

**IOATMaxUsed**

is the highest percentage of the IOAT used since IPL.

**IOATWarning**

is specified by the system programmer in the system configuration file. Message HCP401I is sent when the percentage of IOAT subpool storage in use exceeds this percent.

The percent value for IOATWarning is displayed as -- if STORAGE IOAT is not specified in the system configuration file.

**Permanent**

indicates the number of permanent frames in use by z/VM.

**Reconfigurable**

indicates the number of reconfigurable frames in use by z/VM. Reconfigurable storage can be removed from the z/VM configuration using the SET STORAGE command. If your z/VM system is running on a machine that does not support reconfigurable storage, the reconfigurable value is displayed as Reconfigurable = --.

**Vacating**

indicates the number of frames to be taken offline by the SET STORAGE remove operation in progress. Vacating frames are included in the Reconfigurable count, but not in any Available count. Frames within storage increments already removed from the configuration are not included in this count. When no SET STORAGE remove operation is in progress, the value of this field is 0.

**Frames < 2G****GlobalUnclearedAvail**

indicates the number of frames on the below 2 GB Global Uncleared Avail list, which identifies frames below 2 GB that have not been cleared to 0 and are available for use by CP. There is only one below 2 GB Global Uncleared Avail list, regardless of the number of processors.

**Pageable**

indicates the number of frames below 2 GB that are available for paging.

**LogicalFreeStorage**

indicates the number of frames below 2 GB that are allocated for host logical free storage.

**RealFreeStorage**

indicates the number of frames below 2 GB that are allocated for host real free storage.

**LockedRS**

indicates the number of frames below 2 GB that are locked in host real storage by CP.

**LockedCmd**

indicates the number of frames below 2 GB that are locked in host real storage as a result of the CP LOCK and UNLOCK commands.

**MinidiskCache**

indicates the number of frames below 2 GB that are used for minidisk track cache.

**Nucleus/Prefix**

indicates the number of all identity-mapped frames (host real storage address of the frame = host logical storage address of the system execution space page), including the CP nucleus and prefix pages.

**Trace**

indicates the number of frames in use for trace tables across all processors.

**Other**

indicates the number of all nonpageable CP frames below 2 GB that are not included elsewhere in the response. Other = total number of nonpageable CP frames < 2 GB - (LogicalFreeStorage < 2 GB + RealFreeStorage < 2 GB + Nucleus/Prefix + LockedRS < 2 GB + Trace).

**Frames > 2G****GlobalUnclearedAvail**

indicates the number of frames on the above 2 GB Global Uncleared Avail list, which identifies frames above 2 GB that have not been cleared to 0 and are available for use by CP. There is only one above 2 GB Global Uncleared Avail list, regardless of the number of processors.

**Pageable**

indicates the number of frames above 2 GB that are available for paging.

**LogicalFreeStorage**

indicates the number of frames above 2 GB that are allocated for host logical free storage.

**RealFreeStorage**

indicates the number of frames above 2 GB that are allocated for host real free storage.

**LockedRS**

indicates the number of frames above 2 GB that are locked in host real storage by CP.

**LockedCmd**

indicates the number of frames above 2 GB that are locked in host real storage as a result of the CP LOCK and UNLOCK commands.

**MinidiskCache**

indicates the number of frames above 2 GB that are used for minidisk track cache.

**Other**

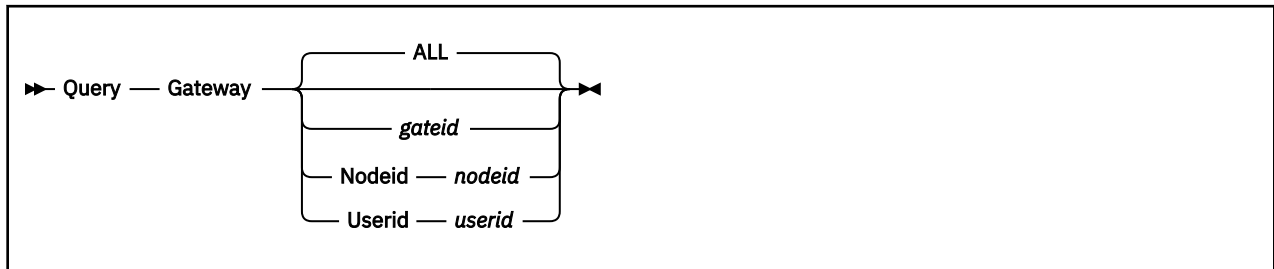
indicates the number of all nonpageable CP frames above 2 GB that are not included elsewhere in the response. Other = total number of nonpageable CP frames > 2 GB - (LogicalFreeStorage > 2 GB + RealFreeStorage > 2 GB + LockedRS > 2 GB). Note that if the total number of nonpageable CP frames > 2 GB = 0, then Other = 0.

### Messages

- HCP263E Too many operands were supplied for this command.



## QUERY GATEWAY



### Authorization

Privilege Class: G

### Purpose

Use the QUERY GATEWAY command to display information about the system and APPC/VM VTAM Support (AVS) gateways defined in the Communication Services (CS) collection. Each node in the CS collection identifies a system gateway that can be used by programs, in the domain of the local VM node, to connect to and from programs in remote nodes.

### Operands

#### ALL

Displays information for all of the defined system gateways and AVS gateways.

#### *gateid*

Displays information for the specified system gateway or AVS gateway.

#### **Nodeid** *nodeid*

Displays information about gateways owned by *nodeid*.

#### **Userid** *userid*

Displays information about gateways owned by *userid* on the local VM node.

### Responses

Response 1:

```
Gateway: gateid Owning Nodeid: nodeid [via tnode]
```

#### *gateid*

Specifies the name of the gateway known in the CS collection.

#### *nodeid*

Specifies the node ID of the system in the CS collection that owns the gateway.

#### *tnode*

Specifies the node through which the *gateid* is known.

Response 2:

```
Gateway: gateid Owning Userid: userid.
```

#### *gateid*

Specifies the name of the gateway defined on the local system.

#### *userid*

Specifies the user ID on the local system that owns the gateway.

**Note:** The owning user ID is displayed only if the gateway resides on the local z/VM system. If a system gateway is defined on the local system, the owning user ID is SYSTEM because the gateway is owned by the local system.

Response 3:

```
Status: status.
```

### ***status***

Specifies either an identify pending or a revoke pending. This response is suppressed if the gateway is identified.

Response 4:

```
No gateways identified.
```

Response 5:

```
No gateways identified by node nodeid.
```

### ***nodeid***

Specifies the node ID that has no gateways identified.

Response 6:

```
No gateways identified by user userid.
```

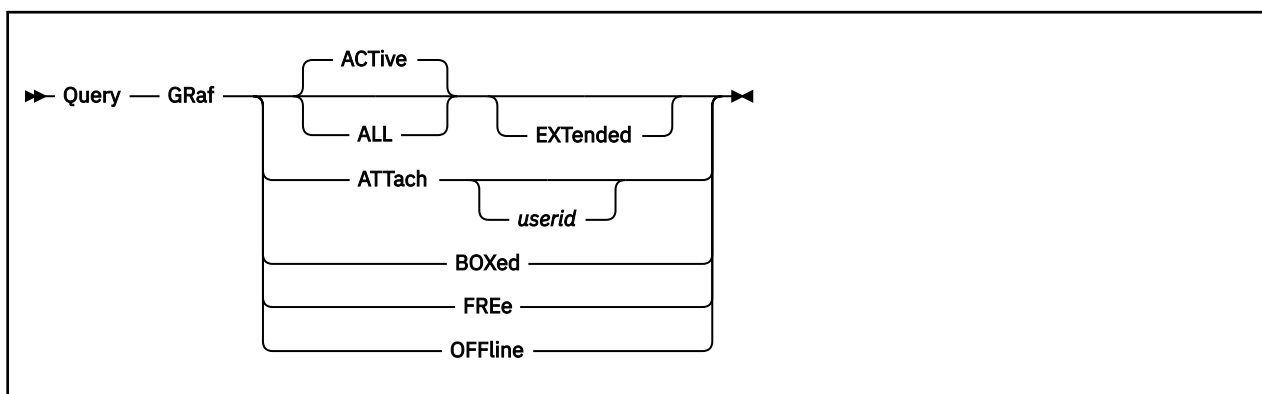
### ***userid***

Specifies the user ID that has no gateways identified.

## **Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP2724E Gateway *gateid* is undefined.

## QUERY GRAF



### Authorization

Privilege Class: B

### Purpose

Use QUERY GRAF to display the status of graphic display devices and 328x display printers.

### Operands

#### ACTIVE

indicates active graphic display devices. An active device is one that is in use by a user or the system and specified as neither FREE nor OFFLINE.

#### ALL

indicates all graphic display devices attached to the system. The status is displayed in the order of ACTIVE, FREE, and OFFLINE. You receive the same response as you would receive if you entered all of the following commands:

```
query graf active
query graf free
query graf offline
```

#### EXTENDED

specifies that logical unit network qualifiers, or logical device host names and IP addresses, are displayed, if they exist.

An IPv4 address is displayed in dotted-decimal format.

**Example:** 9 . 130 . 44 . 144

An IPv6 address is displayed in compressed format (":" designates multiple 16-bit groups of zeros).

**Example:** The following is an IPv6 address:

```
F802:0000:0000:0000:0000:0000:0A00:0001
```

This address is displayed in compressed format as:

```
F802::A00:1
```

#### ATTACH *userid*

indicates the graphic display devices that are dedicated to a user on the system. If you specify *userid*, only displays attached to that user's virtual machine are indicated. If you do not specify *userid*, you

receive the status of all displays dedicated to any user on the system. Attached devices are always active.

**BOXed**

indicates the display devices that are unavailable (usually because of a hardware condition). A boxed display may also be active, free, or offline.

**FREe**

indicates all free display devices. A free display is one that is not currently in use by a user or the system and is not specified as ACTIVE or OFFLINE.

**OFFline**

indicates all offline display devices. An offline display is one specified as neither ACTIVE nor FREE.

**Usage Notes**

1. Within a single VSM that supports network qualified names, multiple responses can occur with the same *luname* when Extended is not specified and one or more VSMS support network qualified names.
2. A VIRTUAL version of this command, QUERY VIRTUAL GRAF, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL GRAF” on page 1303](#).

**Responses**

If the system is using a graphic display device that is in an intervention-required state and you enter QUERY GRAF, you receive a response like this:

```
GRAF 0BDA has intervention required.
```

0BDA is the real device number.

**Note:** This response will be preceded by one of the following.

Response 1:

If you enter a QUERY GRAF, QUERY GRAF ACTIVE, or QUERY GRAF ATTACHED command and a real graphic display is dedicated to a virtual machine, you see something like this:

```
q graf attach
GRAF 0BCD ATTACHED TO OLIVER 0010
Ready;
```

This message means the graphic display device at the real device number 0BCD is attached to OLIVER's virtual machine as the virtual device 0010. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

Response 2:

If you enter QUERY GRAF or QUERY GRAF ACTIVE and a graphic display device has a user logged on or dialed through it, you see something like this:

```
q graf
GRAF 08E0 LOGON AS SVCDIR 0009
VTAM CONTROLS THE FOLLOWING LU'S
GRAF WGU9E316 LOGON AS DONNA 0009
GRAF TN6PV192 LOGON AS CORAK 0009
Ready;
```

This response tells you the user SVCDIR is logged on through the graphic display device at 08E0. The number 0009 is SVCDIR's virtual console device number. The second line of the response tells you the terminals that follow are SNA/CCS terminals, and the VTAM user controls them. VTAM is a VTAM service machine (VSM) recognized by z/VM CP. Any logical units controlled by this VTAM service machine that are logged onto, or dialed into, the system are described in the response following this line. The numbers WGU9E316 and TN6PV192 identify SNA/CCS terminals controlled by a VTAM service machine. They are

the names from the VTAM definition of the terminals as SNA logical units. DONNA and CORAK are the users logged on through VTAM. Each has 0009 as the virtual console device number.

Response 3:

If you enter a QUERY GRAF BOXED, QUERY GRAF FREE, or QUERY GRAF OFFLINE command, you see something like this:

```
q graf free
GRAF 08E1 ENABLED, GRAF 08E2 ENABLED, GRAF 08E3 ENABLED, GRAF 08E4 ENABLED
GRAF 08E5 ENABLED, GRAF 08E6 ENABLED, PRT 08E7 ENABLED, GRAF 0BC0 ENABLED
GRAF 0BC1 ENABLED, GRAF 0BC2 ENABLED, GRAF 0BC3 ENABLED, GRAF 0BC4 ENABLED
GRAF 0BC5 ENABLED, GRAF 0BC6 ENABLED, GRAF 0BC7 ENABLED, GRAF 0BC8 ENABLED
Ready;
```

The word ENABLED might be replaced by OFFLINE, FREE, or BOXED, depending on what you entered.

Response 4:

If you enter a QUERY GRAF ALL command, CP displays all the responses as if you had entered QUERY GRAF ACTIVE, QUERY GRAF FREE, and QUERY GRAF OFFLINE commands.

Response 5:

If CP cannot find a graphic display device of the type you specified, you see a message like this:

```
q graf boxed
A BOXED display was not found.
Ready;
```

In this response, BOXED may be replaced with active, attached, free, or offline, depending on what you specified on the command.

Response 6:

If you enter a QUERY GRAF, QUERY GRAF ACTIVE, or QUERY ALL command with the EXTended option, the response shows the fully qualified logical unit name and might look something like this:

```
Q GR EXT
GRAF 03E0 DIALED TO VTAM 03E0
GRAF 03E1 DIALED TO VTAM 03E1
VTAM CONTROLS THE FOLLOWING LU'S
GRAF VTAM1.LU3E0 LOGON AS CMS0001 03E0
GRAF VTAM1.LU3E1 LOGON AS CLASSG 03E0
```

If instead you enter a QUERY GRAF, QUERY GRAF ACTIVE, or QUERY ALL command without the EXTended option you might see something like this:

```
Q GR
GRAF 03E0 DIALED TO VTAM 03E0
GRAF 03E1 DIALED TO VTAM 03E1
VTAM CONTROLS THE FOLLOWING LU'S
GRAF LU3E0 LOGON AS CMS0001 03E0
GRAF LU3E1 LOGON AS CLASSG 03E0
```

## Messages

- HCP003E Invalid option - *option*
- HCP020E Userid missing or invalid
- HCP263E Too many operands were supplied for this command.

## QUERY HCD

---

►► Query — HCD ◄◄

### Authorization

Privilege Class: B

### Purpose

Use the Query HCD command to display whether HCD is active. If HCD is active, the current production IODF, dynamic hardware change status, software configuration control information, and recovery status will also be displayed.

### Responses

Response 1:

When the Query HCD command is issued and the VM/HCD service virtual machine is not in control of the hardware or software I/O configuration, then the following response is issued:

```
query hcd
HCD is not active
Ready;
```

Response 2:

When the Query HCD command is issued and the VM/HCD service virtual machine is in control of the hardware I/O configuration, but currently is disabled for dynamic hardware changes and is not in control of the software I/O configuration and HCD recovery is not required, then the following response is issued:

```
query hcd
HCD is currently active: IODF = IODF02 PRODIODF
HCD is disabled for dynamic hardware changes
HCD is not controlling the software configuration
HCD recovery is not currently required
Ready;
```

Response 3:

When the Query HCD command is issued and the VM/HCD service virtual machine is in control of the hardware I/O configuration and currently is enabled for dynamic hardware changes, but is not in control of the software I/O configuration and HCD recovery is not required, then the following response is issued:

```
query hcd
HCD is currently active: IODF = IODF02 PRODIODF
HCD is enabled for dynamic hardware changes
HCD is not controlling the software configuration
HCD recovery is not currently required
Ready;
```

Response 4:

When the Query HCD command is issued and the VM/HCD service virtual machine is in control of the hardware I/O configuration and currently enabled for dynamic hardware changes and is also in control of the software I/O configuration and HCD recovery is not required, then the following response is issued:

```
query hcd
HCD is currently active: IODF = IODF02 PRODIODF
HCD is enabled for dynamic hardware changes
HCD is controlling the software configuration: OSCONFIG ID = VMCONFIG
```

```
HCD recovery is not currently required  
Ready;
```

#### Response 5:

When the Query HCD command is issued and the VM/HCD service virtual machine is in control of the hardware I/O configuration but currently disabled for dynamic hardware changes and is also in control of the software I/O configuration, and HCD recovery is required, then the following response is issued:

```
query hcd  
HCD is currently active: IODF = IODF02 PRODIODF  
HCD is disabled for dynamic hardware changes  
HCD is controlling the software configuration: OSCONFIG ID = VMCONFIG  
HCD recovery is currently required  
Ready;
```

### Messages

- HCP003E Invalid option - *option*
- HCP263E Too many operands were supplied for this command.

## QUERY HOLD



### Authorization

Privilege Class: B, D

### Purpose

Use QUERY HOLD to display the following:

- Users in logon hold status
- Users with output in system hold status.

### Operands

#### LOG

displays logon hold information. If you are a class B user, you must specify LOG; this option is restricted to users with class B privileges. The logon hold information is displayed.

### Usage Notes

1. If you are a class B user, you can display logon hold information. If you are a class D user, you can display output hold data. If you hold both privilege classes, you can display either.
2. Users in logon hold status can also have output files in hold status.

### Responses

Response 1:

```
HOLD: {nnnnnnn} PRT, {nnnnnnn} PUN
      {      NO}   {      NO}
```

#### nnnnnnn

#### NO

is the number of files of the type indicated (PRT or PUN), or NO (if none) in hold status. For the class B user, this is the number of files belonging to users in logon hold status. For the class D user, this is the number of files in system hold status. For a user with both classes, it is the combined total.

This response is always the first of two responses issued.

Response 2:

```
NO USERS HELD
```

For the command QUERY HOLD (class D), this means there are no users currently in output hold status. For the command QUERY HOLD LOG (class B), this response means there are no users currently in logon hold status.

Response 3:

```
userid - {ALL},... (up to four users per line)
         {PRT}
         {PUN}
```



is displayed in response to the QUERY HOLD (class D) command for each user in output hold status.

**userid**

identifies the user in output hold status.

**ALL**

**PRT**

**PUN**

indicates which type of output is being held for the user. PRT for printer files only, PUN for punch files only, and ALL for both printer and punch files.

Response 4:

*userid* - LOG, ... (up to four users per line)

is displayed in response to the QUERY HOLD LOG (class B) command for each user in logon hold status.

**userid**

identifies the user in logon hold status.

Response 5:

```
userid - LOG {ALL},...
           {PRT}
           {PUN}
```

is displayed in response to the QUERY HOLD LOG command (Classes B and D) for any user in both logon and output hold status.

**ALL**

**PRT**

**PUN**

indicates in which type of output hold the user may be.

## Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP422E The same option was specified twice.

## QUERY HOTIO

► Query — HOTIO ◄

### Authorization

Privilege Class: B

### Purpose

Use QUERY HOTIO to display the current hot I/O rate settings for devices in the active I/O configuration. (The hot I/O rate is the maximum number of consecutive, unsolicited interrupts per second that CP should allow from an I/O device before refusing to accept input from it.) For information on how to change the hot I/O rate settings for your system, see [“SET HOTIO” on page 1516](#).

### Usage Notes

1. When you enter the QUERY HOTIO command, CP displays the current list of hot I/O rate settings in a form similar to the SET HOTIO command you issued. In some cases, you may see multiple response lines for specific devices because their range and class overlap. When this happens, CP displays the responses in order of precedence: last in, first out.
2. If you want to set the hot I/O rate before the system IPLs, use the HOT\_IO\_RATE statement in the system configuration file. For more information, see [HOT\\_IO\\_RATE Statement in z/VM: CP Planning and Administration](#).

### Responses

If you enter QUERY HOTIO, you receive one of the following responses for each device range and class:

```
HOTIO ON device RATE rate
HOTIO OFF device
```

#### ***device***

indicates the device or devices to which this hot I/O rate setting applies. The variable *device* can be any of the following:

ALL	<i>rdev-rdev</i>	TAPE
DASD	SPECIAL	TERMINAL
GRAF	SWITCH	UR
<i>rdev</i>		

The variables *rdev* and *rdev-rdev* represent a real device number

#### ***rate***

is either:

- A decimal number from 1 to 62500 that indicates the hot I/O rate setting for the specified device or devices
- The word DEFAULT, which indicates a hot I/O rate of 16 for the specified device or devices.

Response 1:

You will see the following response when you do not specify any HOT\_IO\_RATE statements in the system configuration file and you have not issued any SET HOTIO commands:

```
HOTIO ON ALL RATE DEFAULT
```

Note that DEFAULT=16, so all devices in the active I/O configuration have a hot I/O rate of 16.

Response 2:

```
HOTIO ON DASD RATE 00050
```

This response indicates that you either:

- Issued a SET HOTIO DASD 50 command, or
- Specified HOT\_IO\_RATE DASD 50 statement in your system configuration file.

Although they are not listed, all other devices in the active I/O configuration have a hot I/O rate of 16 (DEFAULT). In most of the QUERY HOTIO responses you see, CP will not explicitly tell you that I/O devices are using the default hot I/O rate of 16. There are only two reasons you will see DEFAULT in a QUERY HOTIO response:

1. When no HOT\_IO\_RATE statements or SET HOTIO commands have been processed (as you saw in Response 1).
2. When you use a HOT\_IO\_RATE statement or SET HOTIO command to change the hot I/O rate for one or more devices and you use a second statement or command to change that same device (or devices) back to the default. (You will see an example of this for DASD devices in Response 3.)

Response 3:

This example shows you the responses you would see before and after issuing various SET HOTIO commands.

The following response indicates that all devices in the active I/O configuration are using the default hot I/O rate of 16:

```
q hotio
HOTIO ON ALL RATE DEFAULT
Ready;
```

Here is what you would see if you issued a SET HOTIO DASD command followed by a QUERY HOTIO command:

```
set hotio dasd 25
Ready;

q hotio
HOTIO ON DASD RATE 00025
Ready;
```

Note that you no longer see the response line that indicated all devices were using the default.

Here is what you would see if you issued a SET HOTIO command for a specific range of devices followed by another QUERY HOTIO command:

```
set hotio 280-28f 32
Ready;

q hotio
HOTIO ON 0280-028F RATE 00032
HOTIO ON DASD RATE 00025
Ready;
```

Note that you see the results of both of the SET HOTIO commands you have issued thus far with the last command that you issued as the first response in the list. This response tells you that all DASD devices have a hot I/O rate of 25 and all devices in the range X'0280' to X'028F' have a hot I/O rate of 32. Because the last command you issued is the most recent hot I/O rate change, CP displays that command first. In

## QUERY HOTIO

this example, if any of the devices in the range X'0280' to X'028F' are DASD devices, they will have a hot I/O rate of 32, because your second command overrides the global rate defined in your first command.

Here is what you would see if you issued a SET HOTIO command for a specific device number followed by another QUERY HOTIO command:

```
set hotio 28e 124
Ready;

q hotio
HOTIO ON  028E      RATE 00124
HOTIO ON  0280-028F RATE 00032
HOTIO ON  DASD     RATE 00025
Ready;
```

Note that you see the results of all three SET HOTIO commands you have issued thus far and note that they appear in last in, first out order. This response tells you that all DASD devices have a hot I/O rate of 25 and all devices in the range X'0280' to X'028F' have a hot I/O rate of 32, except for the device at X'028E', which has a hot I/O rate of 124. Again, if any of the devices in the range X'0280' to X'028F' are DASD devices, then they will have a hot I/O rate of 32 (or 124) instead of 25.

Here is what you would see if you issued a SET HOTIO DASD DEFAULT command followed by another QUERY HOTIO command:

```
set hotio dasd default
Ready;

q hotio
HOTIO ON  DASD     RATE DEFAULT
HOTIO ON  028E     RATE 00124
HOTIO ON  0280-028F RATE 00032
Ready;
```

This response tells you that all devices in the range X'0280' to X'028F' have a hot I/O rate of 32 (except for the device at X'028E', which has a hot I/O rate of 124) and all DASD devices have the default hot I/O rate of 16. Note that you issued four SET HOTIO commands and you only see three responses. Two of the commands (the first and the fourth) affecting the same devices (DASD). Because your fourth command completely overrode your first command, CP no longer needs to display the results of your first SET HOTIO DASD command.

Here is what you would see if you issued a SET HOTIO command to turn hot I/O detection off for a specific device followed by a final QUERY HOTIO command:

```
set hotio 28f off
Ready;

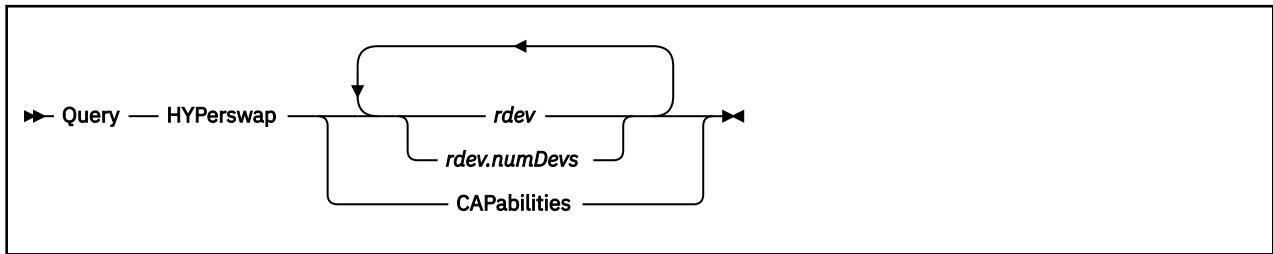
q hotio
HOTIO OFF 028F
HOTIO ON  DASD     RATE DEFAULT
HOTIO ON  028E     RATE 00124
HOTIO ON  0280-028F RATE 00032
Ready;
```

This response tells you that all devices in the range X'0280' to X'028F' have a hot I/O rate of 32 (except for the device at X'028E', which has a hot I/O rate of 124, and device X'028F', which has turned off hot I/O detection) and all DASD devices have the default hot I/O rate of 16. Again, if device X'028F' is a DASD device, then your last SET HOTIO command (which turned off hot I/O detection for device X'028F') will override your previous command (SET HOTIO DASD DEFAULT), causing all DASD devices have the default hot I/O rate of 16, except for the DASD at X'028F' which is no longer detecting interrupts.

## Messages

- HCP002E Invalid operand - *operand*
- HCP6816E Hot I/O rates are being controlled by HCD.

## QUERY HYPERSWAP



### Authorization

Privilege Class: B

### Purpose

Use the QUERY HYPERSWAP command to display the status of a device or to display z/VM capabilities with respect to the HyperSwap function.

### Operands

#### *rdev*

#### *rdev.numDevs*

specifies a single real device (*rdev*) or a real device combined with a decimal range count (*rdev.numDevs*). A single *rdev*, a series (*rdev rdev*), a range (*rdev.numDevs*) or a combination of these forms may be entered. For a device in the active configuration, the device number entered must be a 4-digit hexadecimal number between X'0000' and X'FFFF'. For a device in the standby configuration, the device must be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured. A range of devices cannot span subchannel sets.

#### **CAPabilities**

displays the system's capabilities related to the HyperSwap function.

### Usage Notes

1. Responses are issued for individual devices and appear only for devices that are either quiesced or enabled for automatic quiesce processing. Refer to the HYPERSWAP command (see [“HYPERSWAP” on page 525](#)) for more information.
2. If a device is enabled for automatic quiesce processing and is also currently quiesced, the response indicates that the device is quiesced.
3. Responses to the CAPABILITIES operand are blank delimited tokens which might be in any order. Programs parsing these results must be able to handle tokens in any arbitrary order as well as tolerate undefined tokens.

### Responses

Response 1:

To query the HyperSwap status of devices 190-19F and 200 when the device has been enabled for automatic quiesce, enter the following:

```
Q HYP 190.16 200
Device 0190 is enabled for automatic quiesce processing
  MIH Trigger: Disabled
Device 0191 is quiesced
Device 0192 is enabled for automatic quiesce processing
```

## QUERY HYPERSWAP

```
MIH Trigger: Enabled after 3 missing interrupt(s)
Device 019A is quiesced
Device 019D is enabled for automatic quiesce processing
MIH Trigger: Disabled
Device 019E is enabled for automatic quiesce processing
MIH Trigger: Enabled after 5 missing interrupt(s)
Device 019F is quiesced
Ready;
```

The MIH Trigger line will not be displayed for devices that are already quiesced because the trigger is not relevant at this point.

Response 2:

To query the HyperSwap status of devices 200-207, enter the following:

```
Q HYP 200.8
There are no Hyperswap functions now in effect for the specified devices
Ready;
```

Response 3:

To query the z/VM capabilities related to the HyperSwap function, enter the following:

```
Q HYP CAP
Capabilities: SSI MSS PPRCSN MT OFFSEC
Ready;
```

Response token definitions are:

### SSI

z/VM CP contains support required for a Geographically Dispersed Parallel Sysplex (GDPS) to tolerate a z/VM Single System Image (SSI) in a GDPS HyperSwap environment. Specifically, this pertains to support required by GDPS to automate recovery of GDPS initiated system resets directed to SSI member LPARs.

### MSS

z/VM CP contains support to place secondary Peer-to-Peer Remote Copy (PPRC) DASD in an alternate Subchannel Set for the purpose of being exploited in a GDPS HyperSwap environment.

### MSS4

z/VM CP contains support for 4 subchannel sets limited to only DASD devices in subchannel sets 0, 1, 2 and 3.

### PPRCSN

z/VM CP enables and processes PPRC Suspend Summary Notifications for the purpose of reducing CP host overhead during PPRC state changes, specifically in a GDPS HyperSwap environment.

### MT

z/VM CP contains support for Multi-Target Peer-to-Peer Remote Copy (MT PPRC) DASD.

### OFFSEC

z/VM CP supports the OFFSEC operand on the HYPERSWAP command to vary offline specified additional secondary devices in a Multi-Target PPRC scenario.

### VARYON

z/VM CP contains support for parallel VARY ON of real devices.

## Messages

- HCP002E Invalid operand - *operand*
- HCP6704E Missing token at end of line
- HCP6706E Invalid device address - *address*

## QUERY ICLNAME

```
► Query — ICLNAME — epname ►
```

### Authorization

Privilege Class: A, C, E

### Purpose

Use QUERY ICLNAME to display usage statistics information about indirect calls to external entry points.

### Operands

#### *epname*

is the name of the external entry point for which you want to display information. The variable *epname* must be a 1-character to 8-character string. The first character must be alphabetic or one of the following special characters: dollar sign (\$), number sign (#), underscore (\_), or at sign (@). The rest of the string can be alphanumeric characters, the 4 special characters (\$, #, \_, and @), or any combination thereof.

You can use generic entry point names to request information about a specific subset of entry point names. A generic entry point name is a 1-character to 8-character string with asterisks (\*) in place of 1 or more characters and percent signs (%) in place of exactly 1 character. For example,

```
query iclname hcp%rc*
```

displays all the entry points that start with HCP and have RC as their fifth and sixth characters.

### Usage Notes

1. To display the address of the CP indirect call locator block for a specific exit point, use the LOCATE ICLBK command (see [“LOCATE ICLBK”](#) on page 614).
2. For more information about indirect calls to external entry points, see [Available Diagnostic Facilities in z/VM: CP Exit Customization](#).

### Responses

Response 1:

To display the current usage statistics for all external entry points whose names begin with the letters "RDF", enter the following:

```
query iclname rdf*
ICLNAME      Count Seconds
RDFSAMPL     90 0.000123
RDFEXECU      1 0.000041
Ready;
```

In this example,

#### **ICLNAME**

is the name of the external entry point in the indirect call locator block (ICLBK).

#### **Count**

shows you the total number of successful indirect calls to this external entry point.

### Seconds

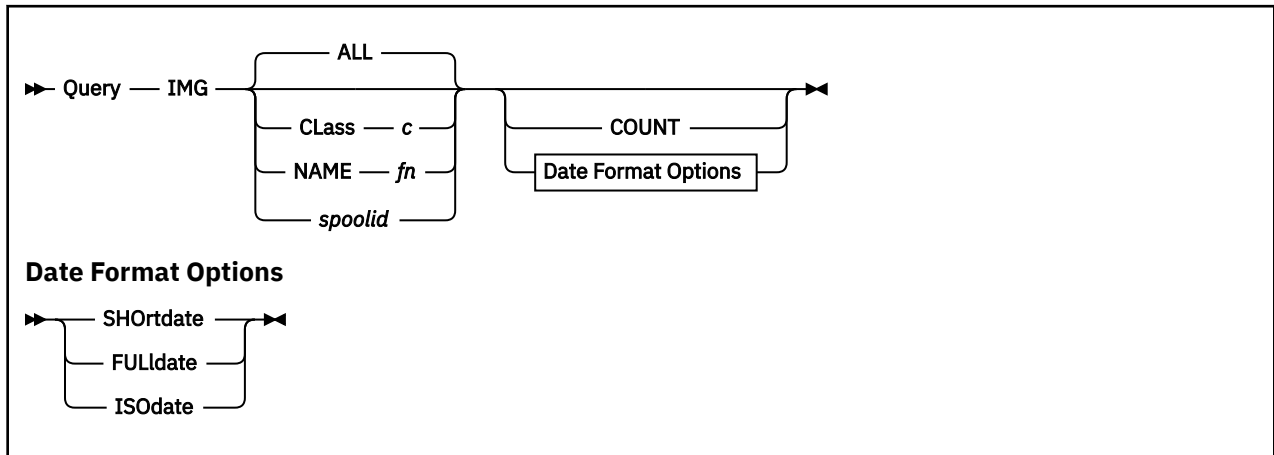
shows the total elapsed time CP used to process all of the successful indirect calls to this external entry point. For each successful indirect call, CP records how long it took (the elapsed time, not CPU time) to process the external entry point from the time it was called to the time it finished processing.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP6706E Invalid entry point name - *epname*
- HCP6720I No items found to satisfy selection criteria.



## QUERY IMG



### Authorization

Privilege Class: A, B, C, D, E

### Purpose

Use QUERY IMG to display information about image library (IMG) files.

### Operands

#### ALL

lets you display information on all files of the specified type. This is the default.

#### Class *c*

specifies the class of files to be queried. The operand *c* is either A or P, where A indicates that the file is in the available state and P indicates that the file is in the pending purge state.

#### NAME *fn*

specifies the file name of the file for which information is displayed.

#### *spoolid*

identifies the file to be queried; this number is assigned by the system.

#### COUNT

displays only the number of image library files that match the criteria specified on the command. No other information is provided.

### Options

#### SHOrtdate

Specifies that dates in the response to this command will be displayed in mm/dd format, where mm is the month and dd is the day of the month.

#### FULldate

Specifies that dates in the response to this command will be displayed in mm/dd/yyyy format, where mm is the month, dd is the day of the month, and yyyy is the 4-digit year.

#### ISOdate

Specifies that dates in the response to this command will be displayed in yyyy-mm-dd format, where yyyy is the 4-digit year, mm is the month, and dd is the day of the month.

## Usage Notes

1. Class G users can use QUERY RDR/PRT/PUN to display comparable information for spool files.
2. If no date format option is specified, then the user's default date format setting is used.

## Responses

Response 1:

The following response is issued when you specify QUERY IMG with the SHORTDATE option.

```
OWNERID  FILE TYPE CL RECS DATE  TIME      FILENAME FILETYPE ORIGINID
*IMG     spid typ  c  nnnn mm/dd hh:mm:ss filename filetype originid
```

Response 2:

The following response is issued when you specify QUERY IMG with the FULLDATE option.

```
OWNERID  FILE TYPE CL RECS DATE      TIME      FILENAME FILETYPE ORIGINID
*IMG     spid typ  c  nnnn mm/dd/yyyy hh:mm:ss filename filetype originid
```

Response 3:

The following response is issued when you specify QUERY IMG with the ISODATE option.

```
OWNERID  FILE TYPE CL RECS DATE      TIME      FILENAME FILETYPE ORIGINID
*IMG     spid typ  c  nnnn yyyy-mm-dd hh:mm:ss filename filetype originid
```

### OWNERID

indicates that the file is an IMG file owned by the system. The designation \*IMG is not a user ID; it indicates the queue on which the file resides.

### FILE

identifies the spool file.

### TYPE

identifies the type of system data file. This is IMG for image library files.

### CL

identifies the file class, either A or P. Class A indicates that the file is available. Class P indicates that the file is in the pending purge state; it is purged during the next system IPL or RESTART.

### RECS

identifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines in thousands rounded to the nearest 1000. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million.

### DATE

identifies the date the file was closed. The format of the date is determined by the SHORTDATE, FULLDATE, or ISODATE options.

### TIME

identifies the time the file was closed, in hours, minutes, and seconds. If the file has not yet been closed, this is the time the file was opened.

### FILENAME

identifies the file name assigned to the file.

### FILETYPE

identifies the file type assigned to the file.

### ORIGINID

identifies the originator of the file.

Response 2:

```
FILES: NO IMG
```

is returned if none of the specified files exists.

Response 3:

If the COUNT option is specified on the QUERY IMG command, only the number of image library files that match the criteria specified is displayed:

```
FILES: {nnnn} IMG
        { NO }
```

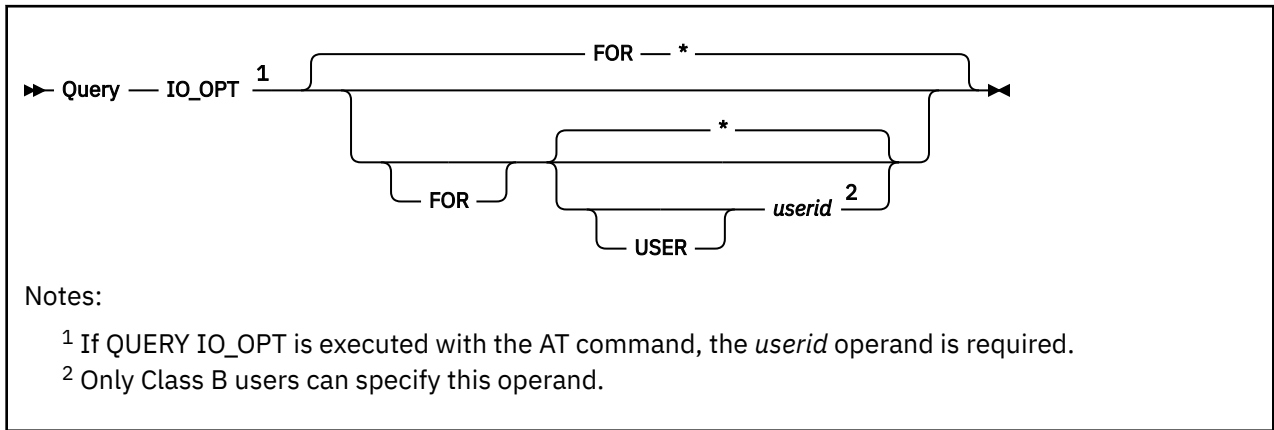
### **nnnn**

identifies the number of image library files. If no files are found that match the selection criteria, NO replaces *nnnn*.

### **Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP849E A user ID must be specified with the spoolid
- HCP1558E Command processing cannot complete.

## QUERY IO\_OPT



### Authorization

Privilege Class: B, G

### Purpose

Use QUERY IO\_OPT to determine the current status of all I/O options for a virtual machine. The setting for I/O options are controlled by the SET IO\_OPT command.

### Operands

#### USER *userid*

\*

specifies an individual user for which the I/O options status is to be reported.

An asterisk (\*), which is the default, specifies the virtual machine (user ID) from which the command is issued.

If QUERY IO\_OPT is executed with the AT command, *userid* operand is required.

### Usage Notes

1. When the setting for all users is OFF, that virtual machine I/O option is disabled for all virtual machines, including any new users that may logon, regardless of the individual setting.  
When the setting for all users is ON, then the individual user setting determines whether I/O options are enabled or disabled for the requested user ID.
2. The default setting for the UID I/O option is to be set ON for all users and each virtual machine. When this option is ON for a virtual machine, two real PCI functions with the same UID value cannot be attached to the same virtual machine.
3. The default setting for the ENFORCE\_BY\_VOLID I/O option is determined by the ENFORCE\_BY\_VOLID configuration statement. When this option is ON, CP enforces attachment of DASD devices by their VOLIDs.

### Responses

Response 1:

If you enter QUERY IO\_OPT with an *userid* or \*, you will see something like:

```
ALL USERS SET IO_OPT UID - {OFF}
                        {ON}
ENFORCE_BY_VOLID - {OFF}
```

```

                                {ON})
OPTION SETTINGS FOR {userid}:
OPTION SETTING
UID      {ON}
        {OFF}

```

The first section of the response displays the setting for all I/O options for all users. The second section of the response displays the settings for all I/O options for the virtual machine specified.

**OPTION**

displays the I/O options which are set to be enabled/disabled.

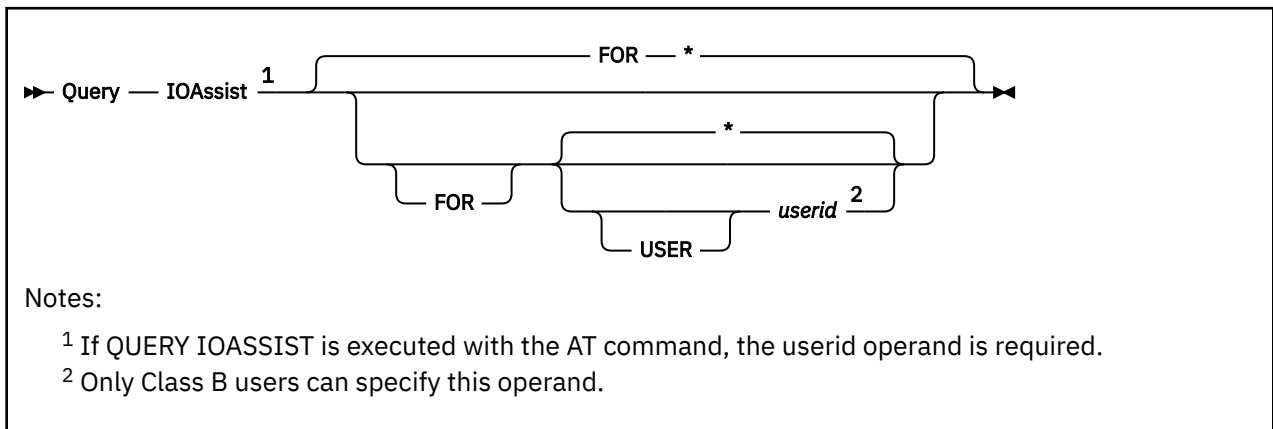
**SETTING**

displays the individual user setting.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command

## QUERY IOASSIST



### Authorization

Privilege Class: B, G

### Purpose

Use QUERY IOASSIST to determine the current status of all I/O assists for a virtual machine. The setting for I/O assists are controlled by the SET IOASSIST command, as well as the SET QIOASSIST command.

### Operands

#### USER *userid*

\*

specifies an individual user for which the I/O assist status is to be reported.

An asterisk (\*), which is the default, specifies the virtual machine (user ID) from which the command is issued.

If QUERY IOASSIST is executed with the AT command, *userid* operand is required.

### Usage Notes

1. When the setting for all users is OFF, eligibility for virtual machine assists is disabled for all virtual machines, including any new users that may logon, regardless of the individual setting.

When the setting for all users is ON, then the individual user setting determines whether I/O assists are allowed or disallowed for the requested user ID.

### Responses

Response 1:

If you enter QUERY IOASSIST with an *userid* or \*, you will see something like:

```
ALL USERS SET IOASSIST - {OFF}
                        {ON}

ASSIST SETTINGS FOR {userid}:

ASSIST SETTING STATUS
ALL      {ON}      {ACTIVE}
         {OFF}    {PENDING ACTIVE}
         {INACTIVE}
         {NOT AVAILABLE}
```

```
{USER DISABLED}
{CP DISABLED}
```

The first line of response displays the setting for all I/O assists for all users. The second line of response displays the userid of the specific virtual machine for which this assist is setting.

**ASSIST**

displays the I/O millicode assists which are set to be enabled/disabled.

**SETTING**

displays the individual user setting.

**STATUS**

displays status for all assists which will each be one of the following:

**ACTIVE**

the specified userid is currently enabled to use the assist.

**PENDING ACTIVE**

the specified userid is in the process of being enabled to use the assist.

**PENDING INACTIVE**

the specified userid is in the process of being disabled from using the assist.

**INACTIVE**

the specified userid is currently not using the assist.

**NOT AVAILABLE**

the assist is not available to the virtual machine.

**USER DISABLED**

the virtual machine was made ineligible to use the assist by the SET IOASSIST OFF or SET QIOASSIST OFF command for the userid.

**CP DISABLED**

the virtual machine cannot use the assist because of special conditions in the virtual machine environment. For example, if the user enabled the tracing of I/O interruptions via the TRACE I/O or TRACE AIF CP commands, the queue-I/O assist is disabled in order to trace the events.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command

## QUERY IOPRIORITY



### Authorization

Privilege Class: A, E

### Purpose

Use QUERY IOPRIORITY to display a user's I/O priority settings or the I/O priority range available to CP.

### Operands

*userid*

\*

specifies the user identification of the virtual machine whose I/O priority range is to be displayed. You can specify an asterisk (\*) to display information about your own virtual machine.

**SYSTEM**

notifies CP to display information about the I/O priority range available to CP.

### Usage Notes

1. The initial I/O priority settings for a user can be specified by the IOPRIORITY statement in the user's directory entry. If an IOPRIORITY statement is not found and LPAR I/O Priority Facility is enabled, LOGON processing requests RELATIVE I/O priority range with a low value of 0 and a high value of 0. The user's effective range is computed based on the I/O priority range available to CP.
2. The I/O priority settings of a user can be changed by the SET IOPRIORITY command. For more information about this command, see [“SET IOPRIORITY” on page 1527](#).

### Responses

Response 1:

The following is the response to QUERY IOPRIORITY SYSTEM:

**LPAR I/O Priority Facility is not installed. Simulated range is 0 to 255**

indicates that the LPAR I/O Priority Facility is not installed, but CP is simulating a range from 0 to 255.

**LPAR I/O Priority Facility is not enabled**

indicates that the LPAR I/O Priority Facility is installed but not enabled for your local partition (LPAR). CP is simulating a range from 0 to 255.

**LPAR I/O Priority Facility is enabled. Range is *nnn* to *mmm***

indicates that the LPAR I/O Priority Facility is installed and enabled for your local partition (LPAR). *nnn* is the lowest I/O priority available to your logical partition and *mmm* is the highest.

Response 2:

One of the following is the response to QUERY IOPRIORITY *userid*:

```

userid  REQUESTED RANGE nnn mmm ABSOLUTE
        EFFECTIVE RANGE xxx yyy-
  
```

or



```
userid  REQUESTED RANGE nnn mmm RELATIVE
        EFFECTIVE RANGE xxx yyy
```

**userid**

is the user ID whose I/O priority settings are being displayed.

**REQUESTED RANGE nnn mmm**

indicates the low and high values of the I/O priority range requested for this user.

**ABSOLUTE****RELATIVE**

indicates the type of range that was requested for this user.

**EFFECTIVE RANGE xxx to yyy**

indicates the low and high values of the I/O priority range that CP will allow this user to use.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*}
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on

## QUERY IPLPARMS

► Query — IPLparms ◄

### Authorization

Privilege Class: A

### Purpose

Use QUERY IPLPARMS to display the initial program load (IPL) parameters currently in CP storage that CP will use for a shutdown re-IPL or a software restart.

### Usage Notes

1. This command displays exactly what was stored in the 240-byte field in CP storage. HCPSAL (the stand-alone loader) passes the IPL parameters to CP and then CP places them into the 240-byte field. You can change the IPL parameters stored in this 240-byte field:
  - a. When creating the Stand-Alone Program Loader (SAPL) using the IPLPARMS operand of the SALIPL utility.
  - b. During initialization, using the panel interface of SAPL.
  - c. After initialization using the SET IPLPARMS command (see [“SET IPLPARMS” on page 1530](#)). The new IPL parameters you specify will then be used during the next shutdown re-IPL or a software restart. They have no effect on the current system.
2. For a description of each of the IPL parameters and more information about SALIPL and SAPL, see [Running Utility Programs in z/VM: System Operation](#).

### Responses

Response 1:

To display the parameters currently stored in the IPL record, enter the following:

```
query iplparms
Current IPL parameters:
FN=SYSTEM      FT=CONFIG      CONS=08E6      PROMPT
Ready;
```

This response tells you that the last time you IPLed, or used the SET IPLPARMS command, you used four IPL parameters to:

- Define the file name of the system configuration file
- Define the file type of the system configuration file
- Specify the address of the operator console
- Indicate that you want CP to prompt you for startup parameters during the next IPL.

CP will use the default values for all the IPL parameters that you did not specify.

Response 2:

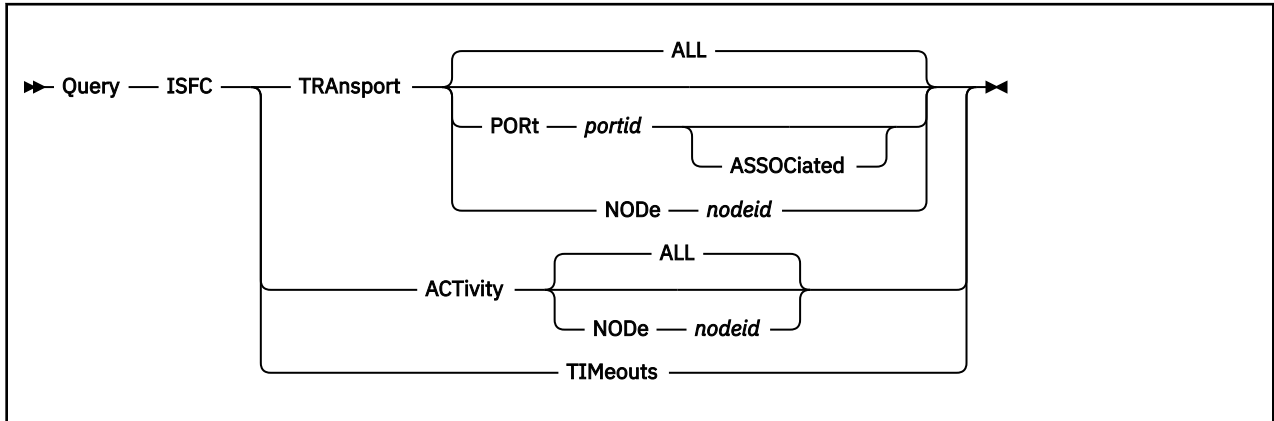
If you did not enter any IPL parameters, you would see the following response:

```
query iplparms  
No IPL parameters are currently defined  
Ready;
```

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*

## QUERY ISFC



### Authorization

Privilege Class: B

### Purpose

Use the QUERY ISFC command to display information about the ISFC Transport end points that have been created or to display logical link information about the logical links to directly connected nodes in the ISFC collection.

### Operands

#### TRANsport

Displays information about the ISFC Transport end points that have been created.

#### ACTivity

Displays logical link information about the logical links to directly connected nodes in the ISFC collection. The information displayed includes message and byte counts.

A logical link is made up of one or more link devices. Use the QUERY ISLINK command to display information about specific link devices.

#### ALL

Displays transport information for all ISFC end points or activity for all nodes. ALL is the default.

#### PORt *portid*

Requests a display of information about the end point matching the specified *portid*.

#### ASSOCiated

Requests a display of information about all end points associated with *portid*.

#### NODe *nodeid*

Requests a display of information about all end points associated with remote node *nodeid*, or all activity associated with remote node *nodeid*.

#### TIMEouts

Displays the status of cross-system timeouts: enabled or disabled. For more information, refer to the CROSS\_SYSTEM\_TIMEOUTS operand on the FEATURES statement in the system configuration file. See [FEATURES statement in z/VM: CP Planning and Administration](#).

### Responses

Response 1:







## QUERY ISLINK

### ***devno1***

Specifies the device number used at *nodeid* to connect to link device *devno*.

Response 2:

```
No links are defined.
```

### **Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP2720E Link device *rdev* not found.
- HCP2720E Node *nodeid* not found.



# QUERY IUCV

Notes:

- <sup>1</sup> Options are valid only for class B users.
- <sup>2</sup> If QUERY IUCV is executed with the AT command, the *userid* or *\*syserv* operand is required.

## Authorization

Privilege Class: B, G

## Purpose

QUERY IUCV displays information about IUCV connections. A class G user can display the user's own IUCV connections. A class B user can display IUCV connection information for any user or IUCV system service.

## Operands

### USER *userid*

identifies a specific user whose IUCV connections are to be displayed.

### \**syserv*

identifies an IUCV system service whose IUCV connections are to be displayed.

## Responses

The command produces a table describing the IUCV paths:

Source	Path	Target	Path	Messages
<i>sourceid</i>	<i>ssss</i>	<i>targetid</i>	<i>tttt</i>	<i>mmmmmmm</i>

where:

### *sourceid*

is the user identifier of the source path owner.

### *ssss*

is the source path ID in hexadecimal.

### *targetid*

is the user identifier of the target path owner.

### *tttt*

is the target path ID in hexadecimal.

### *mmmmmmm*

is the number of outstanding messages on the path.

If \*SCLP connections are displayed, the following response is produced:

Source	Path	Target	Path	Messages	Owner	Class Mask
<i>sourceid</i>	<i>ssss</i>	<i>targetid</i>	<i>tttt</i>	<i>mmmmmmm</i>	<i>ownerid</i>	<i>ccccccc</i>

where:

## QUERY IUCV

**sourceid**

is the user identifier of the source path owner.

**ssss**

is the source path ID in hexadecimal.

**targetid**

is the user identifier of the target path owner.

**tttt**

is the target path ID in hexadecimal.

**mmmmmmmm**

is the number of outstanding messages on the path.

**ownerid**

is the user identifier of the owner of the event class.

**cccccccc**

is the event class mask in hexadecimal.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP045E *userid* not logged on
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*

# QUERY JOURNAL

---

► Query — JOURNAL ◄

## Authorization

Privilege Class: A, E

## Purpose

Use QUERY JOURNAL to determine status of the security journaling function.

## Usage Notes

1. The QUERY JOURNAL command can be used only if the SET\_AND\_QUERY ON operand is specified on the JOURNALING statement in the system configuration file.

Response 1:

```
Journal: LOGON- {on }, LINK- {on }
               {off}, {off}
```

### LOGON ON

### LOGON OFF

Indicates whether accounting records are generated and messages are sent to operations personnel when excessive LOGON attempts with not valid passwords are detected. It also indicates the status of the logon inductor.

### LINK ON

### LINK OFF

Indicates whether type 06 accounting records are generated and messages are sent to operations personnel when excessive LINK attempts with invalid passwords are detected.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

## QUERY KEYALIAS



### Authorization

Privilege Class: G

### Purpose

Use the QUERY KEYALIAS command to display current key label alias definitions. Key label aliases are used on the ATTACH and SET RDEV commands to identify the key label(s) to be used for encrypting data written to a tape.

### Operands

#### ALL

displays the alias name, encoding mechanism, and key label for all defined key aliases. This is the default.

#### *aliasname*

displays the encoding mechanism and key label for the specified alias.

### Responses

Response 1:

Display all defined key label aliases:

```
q keya all
KEYALIAS: (L) DEPTU18
          = TAPE_PVT_U18
KEYALIAS: (L) MYALIAS
          = TAPE_ERF_JLM_LHR_PVT_1024_LBL_01
KEYALIAS: (L) ABC Company
          = E3RW33RSSESYQYPSFTQQPX05399
Ready;
```

Response 2:

Display the definition for the alias "MYALIAS":

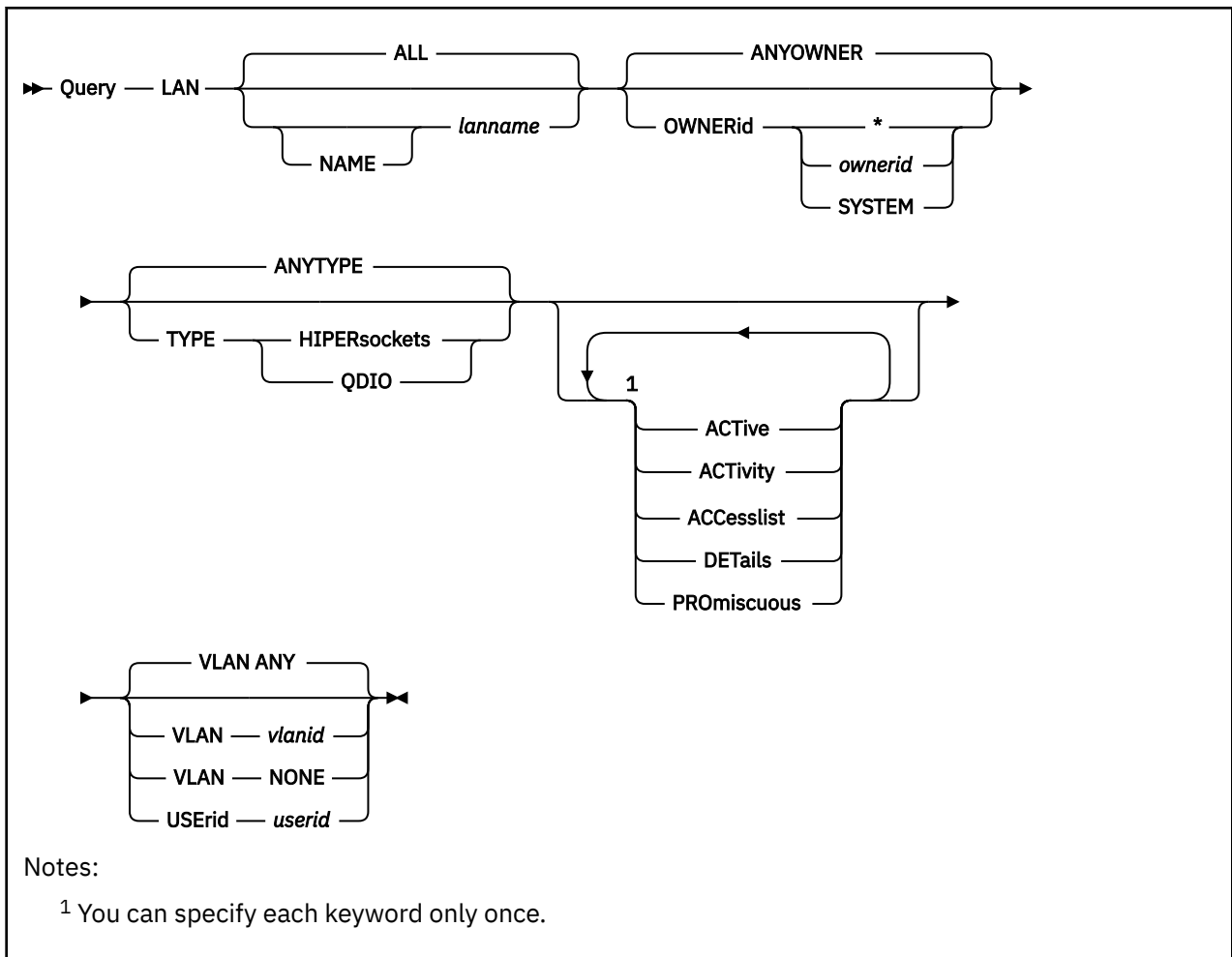
```
q keya myalias
KEYALIAS: (L) MYALIAS
          = TAPE_ERF_JLM_LHR_PVT_1024_LBL_01
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*.
- HCP003E Invalid option - command contains extra option(s) starting with *option*.
- HCP263E Too many operands were supplied for this command.
- HCP6704E Missing token at end of line.
- HCP6706E Invalid string - *string*.

- HCP9968E The key alias does not exist.
- HCP9968I No key aliases found.

## QUERY LAN



### Authorization

Privilege Class: B, G

### Purpose

Use **QUERY LAN** to display information about the designated guest LAN (or any LAN in the System LAN Table). The command keywords and responses documented relate to guest LANs. For information and responses for virtual switches, see [“QUERY VSWITCH” on page 1351](#). When *lanname* represents a virtual switch, use **QUERY VSWITCH** to receive more detailed information.

A Class G user can view any information about any UNRESTRICTED LAN. A Class G user can view information about a RESTRICTED LAN, if the user is the owner of the LAN, in the CP access list for the LAN, or if the user currently has an adapter connected to the LAN.

A Class B user can view information about any LAN in the system.

### Operands

#### ALL

displays information for all possible guest LANs. This is the default when an explicit *lanname* is omitted.

***lanname*****NAME *lanname***

restricts the responses to guest LANs with the specified *lanname*. Both *lanname* and *ownerid* must be known to fully qualify a LAN.

**ANYOWNER**

indicates a request for all qualified guest LANs, regardless of *ownerid* value. This is the default if **OWNERid** is omitted.

**OWNERid \*****OWNERid *ownerid*****OWNERid SYSTEM**

restricts the responses to guest LANs owned by the specified *ownerid*. When **OWNERid \*** is specified, it indicates any LAN owned by the invoker.

**ANYTYPE**

indicates a request for any LAN regardless of LAN TYPE. This is the default when the TYPE operand is omitted.

**TYPE HIPERsockets****TYPE QDIO**

restricts the responses to guest LANs of the specified type.

**ACTive****ACTivity**

displays information about each active connections. Each active connection represents a simulated network interface card (NIC) that has been coupled to the guest LAN. The information displayed includes:

- Identity of the guest LAN.
- The list of users connected to each guest LAN, including the VLAN IDs each is authorized to use.
- IP Addresses registered to the adapter.

**ACcesslist**

displays the current access list for each LAN. For a RESTRICTED LAN, this is the list of all users who are authorized to connect to this LAN with a simulated network adapter. For an UNRESTRICTED LAN, the list will be empty.

This is the list authorized through the use of the SET LAN command using the GRANT and REVOKE operands. If the guest LAN is protected by an External Security Manager (ESM), ESM interfaces must be used to determine the true authorization list.

**DEtails**

displays all available information for each LAN. This is similar to specifying **ACTIVE**, except the display includes users that are coupled, but not active.

**PRomiscuous**

displays a list of users who are authorized for PROMISCUOUS mode on the guest LAN. For an UNRESTRICTED LAN or a HIPERSOCKETS LAN, the list will be empty. No authorization is required for PROMISCUOUS mode with an UNRESTRICTED LAN. PROMISCUOUS mode is not supported on a HIPERSOCKETS LAN.

This is the list authorized through the use of the SET LAN command using the GRANT and REVOKE operands. If the guest LAN is protected by an External Security Manager (ESM), ESM interfaces must be used to determine the true authorization list.

**USERid *userid***

displays the information as requested by other specified options, along with the subset of QUERY LAN DETAILS that applies to the named guest. An error is displayed if *userid* is not coupled.

**VLAN *vlanid***

displays information restricted to connections that have the specified *vlanid* active. **VLAN ANY** displays all connections regardless of any VLAN settings. **VLAN NONE** displays connections that have not enabled VLAN support.

**Examples**

1. QUERY LAN

List information about a specific guest LAN.

```
cp query lan subnta
LAN SYSTEM SUBNTA      Type: QDIO    Connected: 15    Maxconn: INFINITE
PERSISTENT UNRESTRICTED IP      Accounting: ON
IPTimeout: 5           MAC Protection: Unspecified
Isolation Status: OFF    VEPA Status: OFF
Unicast IP address count: 0
```

2. QUERY LAN ACCESSlist:

This example shows the CP access information for LAN SUBNTA.

```
cp query lan subnta accesslist
LAN SYSTEM SUBNTA      Type: QDIO    Connected: 3     Maxconn: INFINITE
PERSISTENT UNRESTRICTED IP      Accounting: ON
IPTimeout: 5           MAC Protection: Unspecified
Isolation Status: OFF    VEPA Status: OFF
Unicast IP address count: 0
```

Note that access is unrestricted, so no list of authorized user IDs is shown.

3. QUERY LAN DETails:

This example displays all the available detail for LAN SUBNTA.

```
cp query lan subnta details
LAN SYSTEM SUBNTA      Type: QDIO    Connected: 3     Maxconn: INFINITE
PERSISTENT UNRESTRICTED IP      Accounting: ON
IPTimeout: 5           MAC Protection: Unspecified
Isolation Status: OFF    VEPA Status: OFF
Unicast IP address count: 4
Adapter Connections:
  Adapter Owner: LINUX5 NIC: 5000.P00 Name: PORT5000 Type: QDIO
  PQUplinkTX: Normal
  RX Packets: 9         Discarded: 0     Errors: 0
  TX Packets: 7         Discarded: 12    Errors: 0
  RX Bytes: 816        TX Bytes: 396
  Device: 5002 Unit: 002 Role: DATA Port: 0079
  Options: Broadcast Multicast IPv6 IPv4 VLAN
  Unicast IP Addresses:
    9.60.59.57         MAC: 02-00-0F-00-00-6E
    FE80::200:F00:100:6E MAC: 02-00-0F-00-00-6E
  Multicast IP Addresses:
    224.0.0.1         MAC: 01-00-5E-00-00-01
    224.0.1.22        MAC: 01-00-5E-00-01-16
    239.255.255.253   MAC: 01-00-5E-7F-FF-FD
    FF02::1           MAC: 33-33-00-00-00-01
    FF02::1:FF00:6E   MAC: 33-33-FF-00-00-6E
  Adapter Owner: TCPIPA NIC: A000.P00 Name: TOSUBNTA Type: QDIO
  PQUplinkTX: Normal
  RX Packets: 124149   Discarded: 0     Errors: 0
  TX Packets: 37595   Discarded: 64    Errors: 0
  RX Bytes: 37792099  TX Bytes: 31343075
  Device: A000 Unit: 000 Role: DATA Port: 0068
  Options: Broadcast Multicast IPv4 VLAN
  Unicast IP Addresses:
    9.60.59.1         MAC: 02-00-0F-00-00-0D
  Multicast IP Addresses:
    224.0.0.1         MAC: 01-00-5E-00-00-01
    224.0.0.9         MAC: 01-00-5E-00-00-09
  Adapter Owner: TCPIP06 NIC: 0600.P00 Name: UNASSIGNED Type: QDIO
  PQUplinkTX: Normal
  RX Packets: 0        Discarded: 0     Errors: 0
  TX Packets: 0        Discarded: 0     Errors: 0
  RX Bytes: 0          TX Bytes: 0
  Device: 0600 Unit: 000 Role: DATA Port: 0134
  Options: Broadcast Multicast IPv4 VLAN
  Unicast IP Addresses:
    9.60.59.6         MAC: 02-00-0F-00-01-24
  Multicast IP Addresses:
    224.0.0.1         MAC: 01-00-5E-00-00-01
```

4. QUERY LAN PROMiscuous:



This example shows the users who are authorized to use promiscuous mode on the simulated network adapter.

```

cp query lan globlan promiscuous
LAN SYSTEM GLOBLAN Type: QDIO Connected: 0 Maxconn: INFINITE
PERSISTENT RESTRICTED IP Accounting: OFF
IPTimeout: 5 MAC Protection: Unspecified
Isolation Status: OFF VEPA Status: OFF
Unicast IP address count: 0
Authorized promiscuous userids:
TCPIPX TCPIP1
    
```

## Responses

Response 1:

```

item ownerid name Type: type Connected: count or Domain: X Maxconn: max
persistence restriction transport MFS: mfsize Accounting: setting
IPTimeout: nnn MAC Protection: setting
Isolation Status: nnn VEPA Status: nnn
Unicast IP address count: nnn
Authorized userids:
user(1) user(2) user(3) user(4) user(5) user(6)
user(7) user(8) user(9) ...
Authorized promiscuous userids:
user(1) user(2) user(3) user(4) user(5) user(6)
user(7) user(8) user(9) ...
Adapter Connections:
Adapter Owner: userid NIC: dev.Pnn Name: portname Type: type
PQUplinkTX: priority
RX packets: rxpackets Discarded: nnn Errors: nnn
TX packets: txpackets Discarded: nnn Errors: nnn
RX bytes: rxbytes TX bytes: txbytes
Device: vdev Unit: nnn Role: role Port: portnum
VLAN: ipversion nnn(1) status
Router: type ipversion vlan
Options: options
Unicast IP Addresses:
ipaddr MAC: macaddr <flags>
ipaddr MAC: macaddr <flags>
Multicast IP Addresses:
mpaddr MAC: macaddr <flags>
mpaddr MAC: macaddr <flags>
    
```

The fields of this response are described as follows (Additional fields may be displayed for virtual switches, see [“QUERY VSWITCH” on page 1351](#)):

**item**

identifies the item being displayed as either a guest LAN or virtual switch (VSWITCH). *ownerid name* uniquely identifies the LAN or VSWITCH.

**Type**

is the LAN type of the adapter (QDIO or HIPERS) or the type of VSWITCH (QDIO or IVL).

**Connected**

*count* indicates the number of coupled adapters.

**Domain: x**

where *x* is the IVL VSWITCH domain and *x* can be A to H.

**Maxconn**

*max* indicates the maximum number of connections (may be INFINITE).

**persistence**

indicates whether the LAN is PERSISTENT or TRANSIENT

**restriction**

indicates whether the LAN is RESTRICTED or UNRESTRICTED.

**transport**

indicates whether the LAN is ETHERNET or IP.

**MFS**

*mfssize* indicates the Maximum Frame Size. It is displayed only for HiperSocket LANs.

**Accounting: ON**

**Accounting: OFF**

indicates whether accounting records will be generated for this LAN.

**user(n)**

represents one user ID in a list of users authorized to COUPLE to this LAN. This is shown for restricted LANs.

**IPTimeout**

indicates the length of time in minutes that a transient IP address table entry remains in the IP address table for the virtual network. A transient entry is an entry added to the virtual network by one guest on behalf of another. This value has no meaning for ETHERNET networks.

**MAC Protection**

*setting* indicates the level of MAC address protection to be used for network devices coupled to the LAN whose MAC address protection settings are not specified (set to UNSPECIFIED) by the SET NIC CP command.

**Isolation Status**

*nnn* indicates the isolation status of the virtual switch and its QDIO data connection.

**Unicast IP address count**

*nnn* indicates the number of Unicast IP addresses associated with this guest LAN.

**Adapter Owner**

*userid* indicates the userid of the virtual machine owning the simulated NIC.

**NIC**

*dev.Pnn* indicates the base virtual device address and port number of the simulated NIC.

**Name**

*portname* indicates the portname assigned to the virtual adapter. The value "UNASSIGNED" is displayed if your software does not assign a value.

**PQUplinkTX**

*priority* indicates the priority with which outbound traffic will be sent to the external network. This is not applicable to guest LANs, so priority for all simulated NICs connected to a guest LAN is Normal.

**RX packets**

*rxpackets* is the number of packets received.

**TX packets**

*txpackets* is the number of packets transmitted.

**Discarded**

*nnn* indicates the number of packets discarded. A packet can be discarded because it is not formatted correctly, for example, or contains an unauthorized VLAN ID. Packets can also be discarded if the LAN is a virtual switch and its internal buffer limit has been reached. The internal buffer limit is reached when the virtual switch is unable to deliver packets to a guest VM port. An example of this is when a guest VM port input buffer is not being serviced quickly enough by the guest operating system. This can happen when a highly utilized guest operating system needs additional resources (memory or processor) to handle the workload.

**Errors**

*nnn* indicates the number of error packets discarded.

**RX bytes**

*rxbytes* is the number of bytes received.

**TX bytes**

*txbytes* is the number of bytes transmitted.

**Device**

*vdev* indicates one of the devices assigned to this connection.

**Unit**

*nnn* indicates the relative device number of this adapter. The base device is unit 000.

**Role**

*role* indicates the role assigned to this device (CTL-READ, CTL-WRITE, or DATA).

**Port**

*portnum* is the port number associated with the guest connection or the virtual switch network connection.

**VLAN**

*ipversion* can be IPv4 or IPv6 when a single VLAN is set per IP Protocol.

*nnnn(n)* indicates the effective VLAN ID or IDs for the device. Multiple VLAN ID lines may be shown.

*status* is "Assigned by system" or "Assigned by user". If the specified VLAN ID conflicts with the authorizations for this user, "Conflicting VLAN ID" is displayed.

**Router**

*type ipversion* indicates the adapter has configured itself as a router with type Primary, Secondary or Multicast in IP version IPv4 or IPv6. *vlan* is either VLAN followed by the VLAN ID, if it is a VLAN-specific router, or Global, if it is a global router. If it is a global router, lines will follow to describe which VLAN IDs the adapter is routing for.

The router line is displayed only when a router is configured.

**Options**

*options* indicates the currently enabled features for this interface, such as Broadcast, Multicast, IPv4 or IPv6.

***ipaddr***

represents one Unicast IP address that has been registered to this device and optional information relating to the IP address.

An IPv4 address is displayed in dotted-decimal format, for example 9 . 130 . 44 . 144.

An IPv6 address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros). For example, the IPv6 address

```
F802:0000:0000:0000:0000:0A00:0001
```

is displayed in compressed format as F802::A00:1.

***mpaddr***

represents one Multicast IP Address.

For IPv4, the address is displayed in dotted-decimal format with the associated MAC address.

For IPv6, the address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros).

**MAC**

*macaddr* represents a MAC address.

***flags***

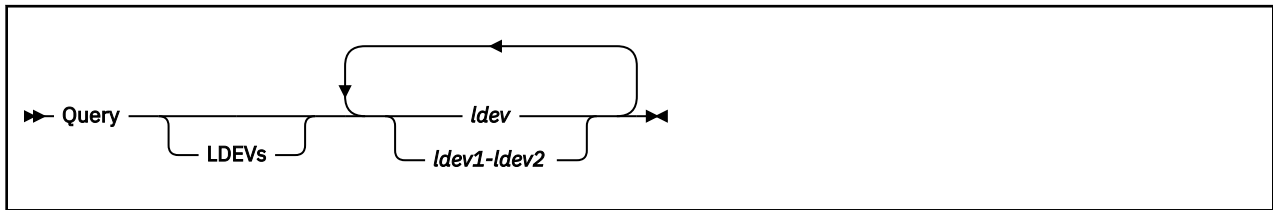
additional information that may be displayed with the IP address. These include:

- Remote indicates that this address was registered automatically to represent a source address found in a datagram. This entry is automatically removed after the IP Timeout (IPTimeout) has expired.
- Manual indicates that this address was registered manually by an ARP cache control command from the driver. This entry will be automatically removed after the IP Timeout (IPTimeout) has expired, or it may be removed earlier by the driver.
- NotOwner means that the IP address was registered by the guest using an ARP Add Unconditional setting, but another guest on the virtual switch had already registered the IP address. If the owner of the IP address removes its registration, this guest can become the owner and receive incoming traffic for the IP address.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP2783E LAN *ownerid lanname* does not exist
- HCP2783E LAN *ownerid lanname* of type *type* does not exist
- HCP2836E VLAN was not found

## QUERY LDEVS



### Authorization

Privilege Class: B, G

### Purpose

Use QUERY LDEVS to display the status of logical devices.

### Operands

#### LDEVs

displays information about logical devices. If you specify LDEVs, the responses show the user ID of the virtual machine that created the logical device.

#### *ldev*

#### *ldev1-ldev2*

displays information about the specified logical devices. Specify each *ldev* as an L followed by 1 to 4 hexadecimal digits. You can specify 1 logical device, a range of logical devices, or any combination thereof.

### Responses

In the following responses, PRT is a printer and GRAF is a display station. Most of the following examples show responses where someone specified LDEVs on the QUERY command. If you omit LDEVs, CP does not display the host information at the end of each response. The rest of the response remains unchanged.

The responses to this query command may be suffixed with one of the following status indicators:

#### 'HOST *hostid*'

usual response - may be followed by 'FROM *ipaddress*' for TCP/IP.

An IPv4 address is displayed in dotted-decimal format.

**Example:** 9 . 130 . 44 . 144

An IPv6 address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros).

**Example:** The following is an IPv6 address:

```
F802:0000:0000:0000:0000:0A00:0001
```

This address is displayed in compressed format as:

```
F802::A00:1
```

#### 'NO HOST'

defined without a *hostid*

#### 'LDBBK DEAD'

LDBBK was flagged as terminated.

## QUERY LDEVS

Response 1:

If a virtual machine user is logged on at the device, you see a response like this:

```
query ldev L101
GRAF L0101 LOGON AS JIM    0009 HOST PVM
Ready;
```

This means that user ID JIM is logged on to the graphic display device at logical device number L0101 and is using virtual console address 0009. PVM is the user ID that created the specified logical device.

Response 2:

If the logical device is dedicated to a user, you see a response like this:

```
query L102
PRT L0102 ATTACHED TO SUSIE  002A
Ready;
```

L0102 is the logical device number of the printer. The printer is attached as the virtual device 002A to user SUSIE's virtual machine.

Response 3:

If the system is using a logical device that needs operator intervention, you see a response like this:

```
query ldev L103
GRAF L0103 has intervention required.
GRAF L0103 LOGON AS LAYLA  03E0 HOST PVM
Ready;
```

L0103 is the logical device number of the graphic display that needs intervention. PVM is the user ID that created the specified logical device.

If the logical device that needs system intervention is a printer, you see a response like this:

```
q L102
Printer L0102 has intervention required.
PRT L0102 ATTACHED TO SUSIE  0102
Ready;
```

Response 4:

If the device is free, you see a response like this:

```
query ldev L102-L105
GRAF L0102 ENABLED  HOST TCPIP    FROM 9.130.58.78
GRAF L0103 ENABLED  HOST PVM
PRT L0104 DISABLED  HOST PVM
GRAF L0105 ENABLED  HOST TCPIP    FROM 9.130.58.29
Ready;
```

PVM and TCPIP are the user IDs that created the specified logical devices.

If you do not specify LDEVS, CP omits the host information at the end of each response and displays instead information on up to 3 logical devices on each line:

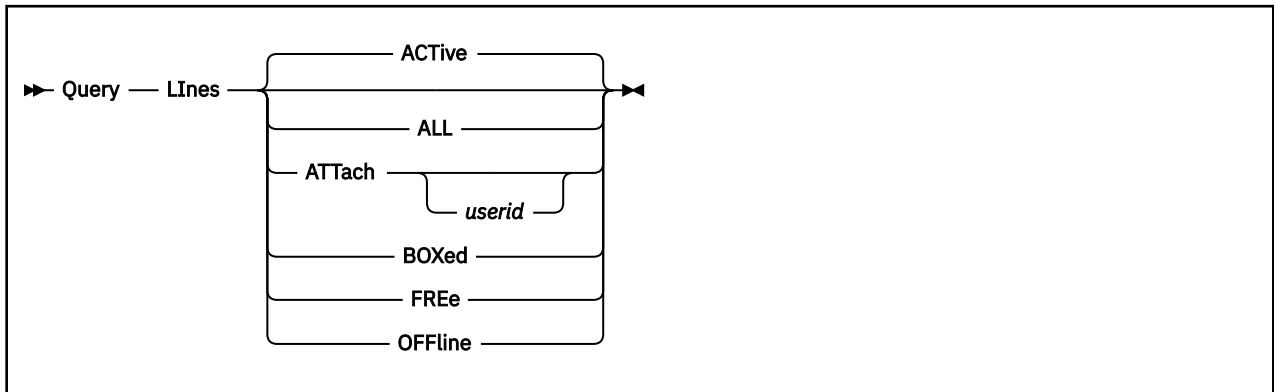
```
query ldevs L0102-L0105
GRAF L0102 ENABLED  , GRAF L0103 ENABLED  , PRT L0104 DISABLED
GRAF L0105 ENABLED
Ready;
```

## Messages

- HCP003E Invalid option - *option*
- HCP009E Invalid range - *range*
- HCP040E Device *ldev* does not exist
- HCP1003E *ldev* was not found.

- HCP1100E A logical device number was not supplied or it is invalid.

## QUERY LINES



### Authorization

Privilege Class: B

### Purpose

Use QUERY LINES to display the status of real communication lines.

### Operands

#### ACTive

indicates active communication lines. An active device is one in use by a user or the system and specified as neither FREE nor OFFLINE.

#### ALL

indicates all communication lines attached to the system. The status is displayed in ACTIVE, FREE, and OFFLINE order and is equivalent to the response you would receive if you entered all of the following commands:

```
query lines active
query lines free
query lines offline
```

#### ATTach *userid*

indicates the communication lines that are dedicated to a user on the system. If you specify *userid*, only lines attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all lines dedicated to any user on the system. Attached devices are always active.

#### BOXed

indicates the communication lines that are unavailable (usually because of a hardware condition). A boxed line may also be active, free, or offline.

#### FREe

indicates all free communication lines. A free line is one that is not currently in use by a user or the system and is specified as neither ACTIVE nor OFFLINE.

#### OFFline

indicates all offline communication lines. An offline communication line is one specified as neither ACTIVE nor FREE.

### Usage Notes

1. The QUERY LINES, QUERY LINKS, and QUERY VIRTUAL LINES commands share the two possible abbreviations "LI" and "LIN". If you are authorized to issue these commands, it is recommended that



you specify at least 4 character abbreviations of the LINES/LINKS operand to ensure that you invoke the intended command. For example, instead of "Q LI" specify "Q LINE".

2. A VIRTUAL version of this command, QUERY VIRTUAL LINES, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see ["QUERY VIRTUAL LINES"](#) on page 1305.

## Responses

If the system is using a communication line that is in an intervention-required state and you enter QUERY LINES, you receive a response like this:

```
LINE 0FE0 has intervention required.
```

0FE0 is the real device number of the communication line that needs the operator's intervention.

**Note:** This response will be preceded by one of the following.

Response 1:

If you enter QUERY LINES, QUERY LINES ACTIVE, or QUERY LINES ATTACHED and a real communication line is dedicated to a user, you see something like this:

```
q lines attach
LINE 00F0 ATTACHED TO OLIVER 0010
Ready;
```

This response means the communication line with the real device number 00F0 is attached to OLIVER's virtual machine as the virtual device 0010. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

Response 2:

If you enter QUERY LINES or QUERY LINES ACTIVE for a communication line that has a user logged on at it or dialed through it, you see something like this:

```
q lines
LINE 00FA LOGON AS DAVID 0009
LINE 00FB DIALED TO PVM 00F9
Ready;
```

This response tells you that the user DAVID is logged on through the communication line at 00FA. The number 0009 is the virtual device number of the virtual console where DAVID is logged on. The second line tells you that the terminal connected to communications line 00FB is dialed to PVM's virtual device number 00F9.

Response 3:

If you enter a QUERY LINES BOXED, QUERY LINES FREE, or QUERY LINES OFFLINE command, you see something like this:

```
q lines free
LINE 00C1 FREE , LINE 00C2 FREE , LINE 00C3 FREE , LINE 00C4 FREE
LINE 00C5 FREE , LINE 00C6 FREE , LINE 00C7 FREE , LINE 00C8 FREE
LINE 00E1 FREE , LINE 00E2 FREE , LINE 00E3 FREE , LINE 00E4 FREE
LINE 00E5 FREE , LINE 00E6 FREE , LINE 00C0 FREE
Ready;
```

The word FREE is replaced by OFFLINE or BOXED, depending on what you entered.

Response 4:

If you enter a QUERY LINES ALL command, CP displays the same response as it would if you had entered QUERY LINES ACTIVE, QUERY LINES FREE, and QUERY LINES OFFLINE commands.

Response 5:

If CP cannot find a communication line of the type you specified, you see a message like this:

## QUERY LINES

```
q lines boxed
A BOXED line was not found.
Ready;
```

BOXED may be replaced with *active*, *attached*, *free*, or *offline*, depending on what you specified on the command.

### Messages

- HCP003E Invalid option - *option*
- HCP020E Userid missing or invalid
- HCP263E Too many operands were supplied for this command.

## QUERY LINKS



### Authorization

Privilege Class: G

### Purpose

Use QUERY LINKS to display link information about a virtual device.

### Operands

#### Links *vdev*

requests the display of information about links to virtual device located at the virtual device number *vdev*.

#### STABLE

requests that QUERY LINKS only display those links that were done with either SR or SW access modes.

### Usage Notes

1. In order to determine if any of the accesses to the specified minidisk are stable accesses, you must issue the command again with the STABLE option.
2. The QUERY LINES, QUERY LINKS, and QUERY VIRTUAL LINES commands share the two possible abbreviations "LI" and "LIN". If you are authorized to issue these commands, it is recommended that you specify at least 4 character abbreviations of the LINES/LINKS operand to ensure that you invoke the intended command. For example, instead of "Q LI" specify "Q LINK".
3. In an SSI cluster where there is sharing of real DASD among multiple z/VM images, QUERY LINKS will display the user IDs of users on the system from which the query is made as well as user IDs of users linked on other systems in the SSI cluster.

### Responses

Response 1:

```

userid vdev {R/W} ,... (up to four responses per line)
            {R/O}
            {R/W ADJUNCT}
            {R/O ADJUNCT}
  
```

#### *userid*

is the user ID of the user who has a link to the specified virtual device.

#### *vdev*

is the virtual device number by which user *userid* refers to the device.

#### R/O

#### R/W

is the type of access user *userid* has to the virtual device. R/O is read-only and R/W is read-write access.

#### ADJUNCT

signifies the virtual device is linked to the named user's adjunct configuration.

## QUERY LINKS

Response 2:

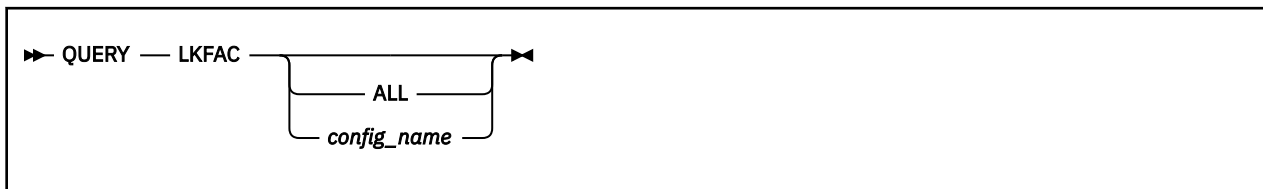
In an SSI cluster, each line of output is prefixed with the system ID and might look similar to this:

```
SYSTEM1 : userid1 0191 R/W, userid2 0191 R/0, userid3 0191 R/0  
SYSTEM2 : userid1 0191 R/0, userid4 0191 R/0
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command.
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP1154I There were no stable links in effect to the specified minidisk.
- HCP6638I LINK information has been truncated.
- HCP6638I No LINK information from other systems in the SSI cluster is included in the response.

# QUERY LKFAC



## Authorization

Privilege Class: G

## Purpose

Use QUERY LKFAC to display information about a Multi-Path Lock Facility RPQ (MPLF) simulation configuration.

## Operands

### ALL

displays the names of all currently defined MPLF configurations.

### config\_name

is the name of the MPLF configuration you want to display.

## Usage Notes

1. If you enter QUERY LKFAC without parameters, CP displays information for the configuration to which the user is currently connected.
2. If you enter QUERY LKFAC with a configuration name, CP displays the information for the specified configuration if the named configuration exists.
3. If you enter QUERY LKFAC ALL, CP displays the names of all defined configurations.

## Responses

Response 1:

```
DEFINED LKFAC CONFIGURATION NAMES:
config_name1 config_name2 config_name3 config_name4
config_name5 .....
```

### config\_name

is the name of the currently defined MPLF configurations.

Response 2:

```
LKFAC config_name  USERS:
                  userid1  useris2  userid3  userid4
                  userid5  userid6  userid7  userid8

I/O Configuration:  SSID      DEVICE RANGE
                   ssss      vdev-vdev
                   ssss      vdev-vdev
                   ssss      vdev-vdev
                   .
                   .
                   .
                   ssss      vdev-vdev
```

## QUERY LKFAC

### **config\_name**

is the name of the MPLF configuration being displayed.

### **userid**

is the user ID of the connected user or users.

### **ssss**

is the virtual subsystem ID (SSID) assigned.

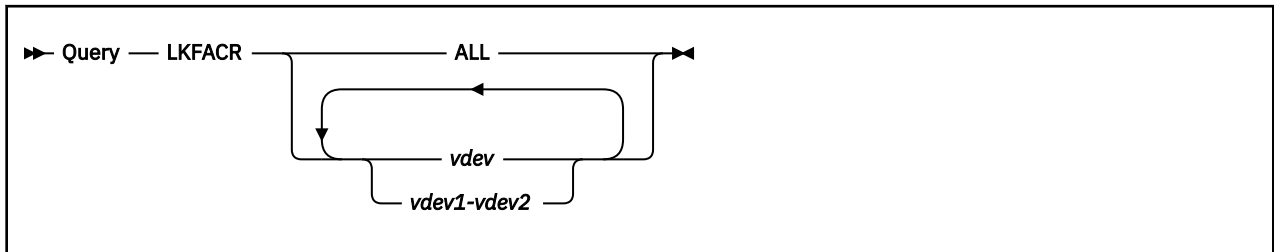
### **vdev**

is the virtual device number ranges associated with the SSID.

## **Messages**

- HCP9961E LKFAC configuration *config\_name* cannot be found
- HCP9962E *userid* not attached to a LKFAC configuration
- HCP9963I No LKFAC configurations defined

## QUERY LKFACR



### Authorization

Privilege Class: G

### Purpose

Use QUERY LKFACR to determine which full-pack minidisks are using the real Multi-Path Lock Facility (MPLF).

### Operands

#### ALL

indicates that the query will list all full-pack minidisks and all dedicated DASD in the virtual configuration that are enabled for real MPLF.

#### *vdev*

#### *rdev1-rdev2*

is the virtual device number or range of virtual device numbers of the devices to be queried for real MPLF.

### Responses

Response 1:

```
Real MPLF is {enabled} for DASD vdev
             {disabled}
```

is issued when the command completes successfully.

Response 2:

```
Real MPLF is enabled for the following full-pack
minidisks: vdev1 vdev2 ... vdevn
```

```
Real MPLF is always enabled for the following dedicated
devices: vdev1 vdev2 ... vdevn
```

is issued when the command completes successfully, and at least one device in each response is enabled for real MPLF. All eligible devices that are part of the queried range are listed in their respective responses. Either response is suppressed if there are no eligible devices for its list.

Response 3:

```
Real MPLF is not available for any virtual devices
in your configuration
```

is issued when the command completes successfully, and no devices from the queried range are enabled for real MPLF.

Response 4:

```
Real MPLF is always enabled for dedicated DASD vdev
```

is issued when the target device is dedicated.

Response 5:

```
Real MPLF is always disabled for non-full-pack  
minidisk vdev
```

is issued when the target device is a non-full-pack minidisk.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP040E Device {*vdev*} does not exist
- HCP6706E Invalid address range - *range*
- HCP9966E You do not have LKFAC directory authorization
- HCP9967E *type vdev* LKFACR cannot be used - no real MPLF is available



# QUERY LOADDEV

► Query — LOADDEV ◄

## Authorization

Privilege Class: G

## Purpose

Use the **QUERY LOADDEV** command to display LOADDEV parameters for a guest IPL. The parameters are for a list-directed IPL without dump from a SCSI or ECKD device. The parameters can identify the operating system program and information to pass to the program. The parameters can also identify the virtual device number and specify whether the IPL is a secure IPL.

## Responses

The following lines are displayed if LOADDEV parameters are set for a SCSI device:

```
SCSI    DEVICE {TBD } {NOSECURE}
             {ddd} {SECURE  }
PORTNAME hhhhhhhh hhhhhhhh          LUN hhhhhhhh hhhhhhhh          BOOTPROG {n }
                                       {AUTO}
BR_LBA  hhhhhhhh hhhhhhhh
```

The following lines are displayed if LOADDEV parameters are set for an ECKD device:

```
ECKD    DEVICE {TBD } {NOSECURE}
             {ddd} {SECURE  }
BOOTPROG {n } BOOTREC {cylinder head record}
          {AUTO}       {LABEL }
```

The following lines are displayed if hex SCPDATA is set:

```
SCPDATA
00----03 04----07 08----11 12----15 16----19
0000 hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
0020 hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
0040 hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
.
.
```

The following lines are displayed if text SCPDATA is set:

```
SCPDATA
0-----1-----2-----3-----4-----
0000 ttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
0050 ttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
0100 ttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
.
.
```

## SCSI

The SCSI response indicates that LOADDEV parameters are set for the IPL of a SCSI device. SCSI is the default device type.

## ECKD

The ECKD response indicates that LOADDEV parameters are set for the IPL of an ECKD device.

**DEVICE TBD**

**DEVICE *dddd***

The DEVICE TBD response indicates that the LOADDEV parameters do not include the device number. The device number is to be determined later.

If the SECURE parameter is not set, then the device number can be specified later as an operand of the **IPL** command. If the SECURE parameter is set, then the device number must be set by using the **SET LOADDEV** command.

The DEVICE *dddd* response indicates the 4-digit hexadecimal virtual device number (*dddd*) of the device to be IPLed.

There is no default value for the DEVICE parameter.

**PORTNAME *hhhhhhhh hhhhhhhh***

The PORTNAME response indicates the hexadecimal 8-byte fibre channel port name of the SCSI device. The default value is 0.

**LUN *hhhhhhhh hhhhhhhh***

The LUN response indicates the hexadecimal 8-byte logical unit number of the SCSI device. The default value is 0.

**BOOTPROG *n***

**BOOTPROG AUTO**

The BOOTPROG response indicates the boot program that is loaded from the IPL device. The *n* value indicates the program number. The program number must be a decimal value in the range 0 - 30. The BOOTPROG AUTO response indicates that the first operating system (not dump program) on the IPL device is loaded. The default value is BOOTPROG 0.

**BR\_LBA *hhhhhhhh hhhhhhhh***

The BR\_LBA response indicates the hexadecimal logical-block address of the boot record of the SCSI device. The default value is 0.

**BOOTREC *cylinder head record***

**BOOTREC LABEL**

The BOOTREC response indicates the location of the boot record. The location can be a specification of cylinder, head, and record on the IPL device. The values are delimited by blanks and contain no leading zeros. The values must be in hexadecimal format.

A BOOTREC LABEL response indicates that the location of the boot record is determined from the volume label. BOOTREC LABEL is the default setting.

**SECURE**

**NOSECURE**

The SECURE response indicates that the SECURE parameter is set. The NOSECURE response indicates that the SECURE parameter is not set. The SECURE parameter must be set before you can initiate a secure IPL. NOSECURE is the default setting.

**SCPDATA**

The SCPDATA response indicates the information that is passed to the IPL program. The information is displayed in the same format (text or hex) as it was entered.

**Usage Notes**

- Parameters for a list-directed IPL without dump (also called LOADDEV parameters) are set by using any of the following methods:
  - **SET LOADDEV** command.
  - LOADDEV user directory statement. The LOADDEV user directory statement can be modified by a directory management tool such as IBM Directory Maintenance (DirMaint). To set load parameters by using DirMaint, use the **DIRM LOADDEV** command. The LOADDEV user directory statement can be modified also by the Image\_IPL\_Characteristics\_Define\_DM API function.

Parameters for a list-directed IPL **with** dump (also called DUMPDEV parameters) are set by using the CP **SET DUMPDEV** command.

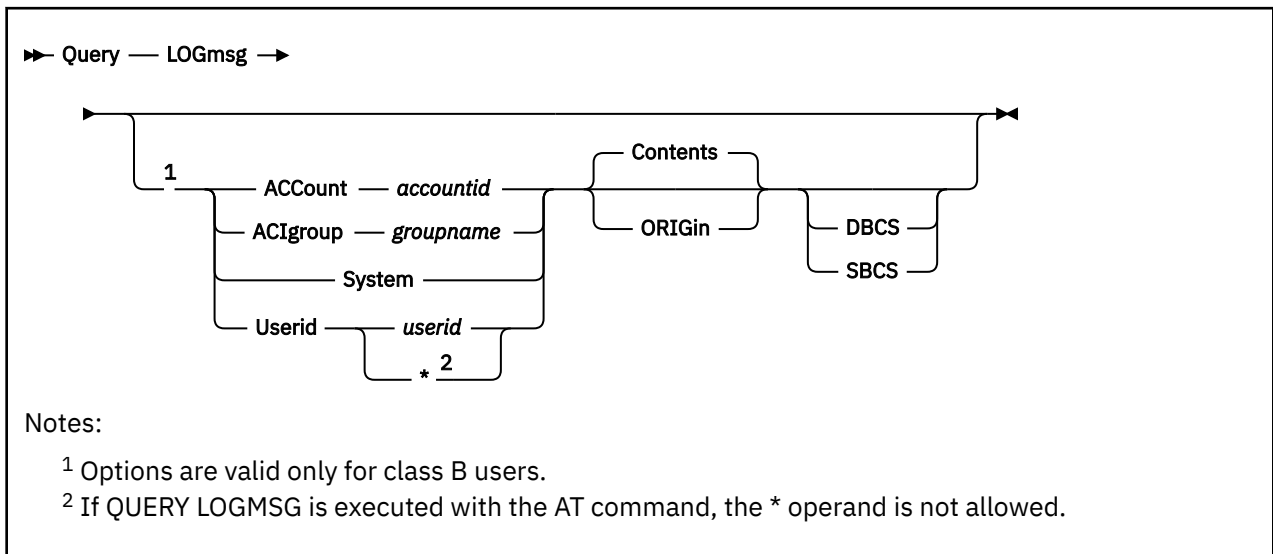
For more information, see [SET LOADDEV](#) and [SET DUMPDEV](#), [LOADDEV Directory Statement](#) in *z/VM: CP Planning and Administration*, [DIRM LOADDEV](#) in *z/VM: Directory Maintenance Facility Commands Reference*, and [Image\\_IPL\\_Characteristics\\_Define\\_DM](#) API function in *z/VM: Systems Management Application Programming*.

2. If a LOADDEV parameter is not set and has a default value, then the **QUERY LOADDEV** response displays the default parameter value.
3. The default LOADDEV and DUMPDEV values for the PORTNAME and LUN parameters is 0, which is not a valid device port name and not a valid device LUN. An IPL attempt that uses the default value for the PORTNAME or LUN parameter will fail.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP2824I No LOADDEV parameters are currently defined

## QUERY LOGMSG



### Authorization

Privilege Class: A, B, C, D, E, F, G

### Purpose

Use QUERY LOGMSG to display the system log messages. The log messages can reside in either a CMS file or the checkpoint area.

If the FEATURES ENABLE LOGMSG\_FROM\_FILE statement is specified in the system configuration file, CP reads the CMS files on the CP-accessed minidisks to determine which log message files to display, and the QUERY LOGMSG command options are valid for a class B user. For more information on the FEATURES statement, see *z/VM: CP Planning and Administration*.

If the FEATURES ENABLE LOGMSG\_FROM\_FILE statement is not specified in the system configuration file, CP displays the log message saved by the SET LOGMSG command.

### Operands

#### ACCount *accountid*

tells CP to display either the contents or origin information of the file that only users whose account ID (set in the user's directory entry) matches *accountid* would see. If you are using a non-DBCS terminal and have not specified the DBCS option, CP searches for a file called *accountid* LOGMSACC on the lowest accessed CP disk. If you are using a DBCS terminal or have specified the DBCS option, CP searches for a file called *accountid* LOGDBACC on the lowest accessed CP disk. If no such file exists, CP searches for a file called *accountid* LOGMSACC.

#### ACIgroup *groupname*

tells CP to display either the contents or origin information of the file that only users who are in the ACI group identified by *groupname* would see. If you are using a non-DBCS terminal and have not specified the DBCS option, CP searches for a file called *groupname* LOGMSACI on the lowest accessed CP disk. If you are using a DBCS terminal or have specified the DBCS option, CP searches for a file called *groupname* LOGDBACI on the lowest accessed CP disk. If no such file exists, CP searches for a file called *groupname* LOGMSACI.

#### System

tells CP to display either the contents or origin information of the file that all users of the system would see. If you are using a non-DBCS terminal and have not specified the DBCS option, CP searches

for a file called SYSTEM LOGMSG on the lowest accessed CP disk. If you are using a DBCS terminal or have specified the DBCS option, CP searches for a file called SYSTEM LOGDBCS on the lowest accessed CP disk. If no such file exists, CP searches for a file called SYSTEM LOGMSG.

### **Userid *userid***

tells CP to display either the contents or origin information of the file that only the specified *userid* would see. If you are using a non-DBCS terminal or have not specified the DBCS option, CP searches for a file called *userid* LOGMSUSR on the lowest accessed CP disk. If you are using a DBCS terminal or have specified the DBCS option, CP searches for a file called *userid* LOGDBUSR on the lowest accessed CP disk. If no such file exists, CP searches for a file called *userid* LOGMSUSR.

### **Userid \***

tells CP to display either the contents or origin information of the file that only your user ID would see. If you are using a non-DBCS terminal or have not specified the DBCS option, CP searches for a file called *userid* LOGMSUSR on the lowest accessed CP disk. If you are using a DBCS terminal or have specified the DBCS option, CP searches for a file called *userid* LOGDBUSR on the lowest accessed CP disk. If no such file exists, CP searches for a file called *userid* LOGMSUSR. In either case, *userid* is the user ID of the command issuer.

### **Contents**

tells CP you wish to see the contents of a file. This is the default.

### **ORIGIN**

tells CP you wish to see on which CP-accessed minidisk the file resides instead of seeing the contents of the selected file.

### **DBCS**

tells CP you wish to see origin information or content for the file that would be displayed to users on DBCS-capable terminals. If neither DBCS nor SBCS are specified, CP assumes you wish to see origin information or content for the file that would be displayed to users on terminals similar to your own.

### **SBCS**

tells CP you wish to see origin information or content for the file that would be displayed to users on SBCS-capable terminals. If neither DBCS nor SBCS are specified, CP assumes you wish to see origin information or content for the file that would be displayed to users on terminals similar to your own.

## **Usage Notes**

1. If the file-based LOGMSG support is not enabled through the FEATURES statement in the system configuration file, QUERY LOGMSG cannot be entered with additional operands.
2. If you enable the file-based LOGMSG support but you do not also enable the LOGMSG support for account IDs, ACI groups, and user IDs, CP searches only for a file called SYSTEM LOGMSG (or SYSTEM LOGDBCS if you are on a DBCS terminal).
3. If you enter the QUERY LOGMSG command with no additional operands, you have enabled the file-based LOGMSG support, and you have also enabled LOGMSG support based on account IDs, ACI groups, or user IDs, CP displays the contents of the appropriate files in the following order:
  - System log message
  - Account log message
  - ACI group log message
  - User log message.

You will be unable to determine where the contents of one file ends and the contents of the next file begins by looking at the response to the QUERY LOGMSG command.

4. If you are a class G user and you enter the QUERY LOGMSG command, you see a log message containing information seen by everyone, information seen only by other users that have your account ID, information seen only by other users that are in the same ACI group as you, and information seen only by you.

5. When you log on, CP processes a special QUERY LOGMSG on your behalf. Only records beginning with an asterisk (\*) are displayed in response to this command. If you enter a QUERY LOGMSG, all records in the log message are displayed.
6. To change the contents of the log message, you must first determine whether you have enabled the file-based LOGMSG support. If you have enabled the file-based LOGMSG support, you must edit the files containing the log messages and enter the CPACCESS command to have CP access the changed files. If you have not enabled the file-based LOGMSG support, you must use the SET LOGMSG command to change the contents of individual lines in the log message.

### Responses

If there is no log message data available, you get this response:

```
There is no logmsg data
```

Otherwise, the contents of the log message, preceded by the LOGMSG header, are displayed on your terminal.

If old-style log message support is in effect, class B users see each line of log message data preceded by a line number. These numbers allow class B users to edit the log message using the class B SET LOGMSG command.

When you specify ORIGIN on the command, you get this type of response:

```
LOGMSG file SYSTEM LOGMSG located on CP accessed disk A (MAINT's 300).
```

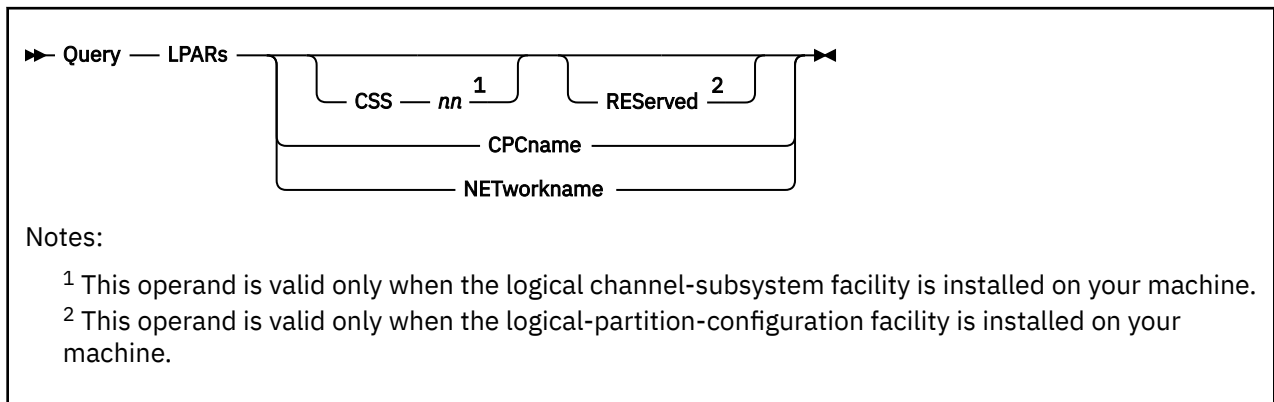
If the requested file does not exist, you see this:

```
LOGMSG file MAINT LOGMSUSR not located on any CP-accessed disk.
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP007E Invalid userid - *userid*
- HCP026E Operand missing or invalid
- HCP041E LOGMSG *nn* does not exist.
- HCP1506E The last line (50) in the log message is already in use.
- HCP6704E Missing token at end of line
- HCP6706E Invalid {group name|account ID} - {*groupname*|*accountid*}
- HCP6734E Support for file-based LOGMSG is not enabled.
- HCP6771E LOGMSG support based on {Account ID|ACI group|Userid} is not enabled.

## QUERY LPARS



### Authorization

Privilege Class: B

### Purpose

Use QUERY LPARS to list all the logical partitions (LPARs) defined on your machine and to display which LPAR is currently active, to list all of the LPARs that have access to a particular channel-subsystem image, to list all of the reserved LPARs in the system or in a particular channel-subsystem image, to display the network name associated with your processor, or to display the CPC name associated with your processor.

### Operands

#### CSS *nn*

is the channel-subsystem ID for which you want partition access information. The variable *nn* must be a 1- to 2-digit hexadecimal number between X'00' and X'0F'. Your machine is likely to support less than 16 channel-subsystem images.

**Note:** This operand requires that the logical channel-subsystem facility be installed.

#### REServed

indicates that only reserved partition slots be displayed.

**Note:** This operand requires that the logical-partition-configuration facility be installed.

#### CPCname

indicates that the CPC name associated with your processor should be displayed.

#### NETworkname

indicates that the network name associated with your processor should be displayed.

### Usage Notes

1. If your system is not enabled for dynamic I/O capabilities, then this command will not complete successfully. However, it will issue a response that indicates the name of the logical partition where your system is running.
2. For more information about logical partitions (LPARs), see *z/VM: CP Planning and Administration*.
3. The network name and CPC name that can be displayed by this command are user-defined names that are created and modified by the Hardware Management Console (HMC) and the Service Element (SE).

### Responses

Response 1 (non-LCSS system):

The response appears as follows when there is no logical channel-subsystem facility installed.

```
query lpars
Partitions on your machine:
Active partition: ZVMLPAR1 Partition number: 1
Partition name:   ZVMLPAR2 Partition number: 2
Partition name:   ZOSLPAR1 Partition number: 3
Partition name:   ZOSLPAR2 Partition number: 4
Ready;
```

This response tells you that you have 4 partitions and that the partition processing this command is ZVMLPAR1.

Response 2 (LCSS system):

The response appears as follows when the logical-channel-subsystem facility is installed and there are no reserved partition slots.

```
query lpars
Partitions on your machine:
Active partition: ZVMLPAR1 CSS: 0 MIF-ID: 1
Partition name:   ZOSLPAR1 CSS: 0 MIF-ID: 2
Partition name:   ZVMLPAR2 CSS: 1 MIF-ID: 1
Partition name:   ZOSLPAR2 CSS: 1 MIF-ID: 2
Ready;
```

This response tells you that you have 4 partitions, two that have access to channel-subsystem image 0 and two that have access to channel-subsystem image 1. The partition processing this command is ZVMLPAR1.

Response 3 (LCSS system with reserved partitions):

The response appears as follows when both the logical-channel-subsystem facility and the logical-partition-configuration facility are installed, and there are reserved partition slots defined.

```
query lpars
Partitions on your machine:
Active partition: ZVMLPAR1 CSS: 0 MIF-ID: 1
Partition name:   ZOSLPAR1 CSS: 0 MIF-ID: 2
Reserved:         *      CSS: 0 MIF-ID: 4
Partition name:   ZVMLPAR2 CSS: 1 MIF-ID: 1
Partition name:   ZOSLPAR2 CSS: 1 MIF-ID: 2
Reserved:         *      CSS: 1 MIF-ID: 3
Ready;
```

This response tells you that you have 4 partitions, two that have access to channel-subsystem image 0 and two that have access to channel-subsystem image 1. The response also tells you that you have two reserved partition slots, one available in channel-subsystem image 0 and one available in channel-subsystem image 1. The partition processing this command is ZVMLPAR1.

Response 4 (Query for a defined CSS):

To display information about the logical partitions that have access to channel-subsystem image 0 when the logical channel-subsystem facility is installed, enter the following:

```
query lpars css 0
Partitions with access to CSS 0:
Active partition: ZVMLPAR1 CSS: 0 MIF-ID: 1
Partition name:   ZOSLPAR1 CSS: 0 MIF-ID: 2
Ready;
```

This response tells you that you have 2 partitions that have access to channel-subsystem image 0. The partition processing this command is ZVMLPAR1.

Response 5 (Query for a defined CSS with reserved partitions):

To display information about the logical partitions that have access to or are reserved in channel-subsystem image 0 when the logical-channel-subsystem facility and the logical-partition-configuration facility are installed, enter the following:



```

query lpars css 0
Partitions with access to CSS 0:
Active partition: ZVMLPAR1  CSS: 0  MIF-ID: 1
Partition name:  ZOSLPAR1  CSS: 0  MIF-ID: 2
Reserved:        *         CSS: 0  MIF-ID: 4
Ready;

```

This response tells you that you have 2 partitions that have access to channel-subsystem image 0 and that you have 1 reserved partition slot available in channel-subsystem image 0. The partition processing this command is ZVMLPAR1.

Response 6: (Reserved partitions in LCSS system):

To display information about the reserved logical partitions on your machine, enter the following:

```

query lpars reserved
Reserved partitions on your machine:
Reserved:        *         CSS: 0  MIF-ID: 4
Reserved:        *         CSS: 1  MIF-ID: 3
Ready;

```

This response tells you that you have 2 reserved partition slots available on your machine, one in channel-subsystem image 0 and one in channel-subsystem image 1.

Response 7: (Reserved partitions in a CSS):

To display information about the reserved logical partitions in a channel-subsystem image, enter the following:

```

query lpars css 1 reserved
Reserved partitions in CSS1:
Reserved:        *         CSS: 1  MIF-ID: 3
Ready;

```

This response tells you that you have 1 reserved partitions slot available in channel-subsystem image 1.

Response 8 (Query for an undefined CSS):

If no logical partitions have access to the specified channel-subsystem image, the response appears as follows.

```

query lpars css 5
There are no partitions that have access to CSS 5.
Ready;

```

Response 9 (Query when no reserved partitions in system):

If no reserved logical partition slots are available on your machine, the response appears as follows.

```

query lpars reserved
There are no reserved partitions available on your machine.
Ready;

```

Response 10 (Query when no reserved partitions in CSS):

If no reserved logical partition slots are available in the specified channel-subsystem image, the response appears as follows.

```

query lpars css 5 reserved
There are no reserved partitions in CSS 5.
Ready;

```

Response 12 (No LPAR Data - Dynamic I/O not enabled):

If your logical partition is not enabled for dynamic I/O changes, the response appears as follows.

```

query lpars
No LPAR data is available.
Active partition: ZVMLPAR1
Ready;

```

## QUERY LPARS

Response 13 (CPC name):

To display the CPC name associated with your processor, enter the following:

```
query lpars cpcname
The CPC name for this processor is FSISKIS9.
Ready;
```

Response 14 (No CPC name):

If there is no CPC name associated with your processor, you would see:

```
query lpars cpcname
There is no CPC name assigned to this processor.
Ready;
```

Response 15 (Network name):

To display the network name associated with your processor, enter the following:

```
query lpars network
The network name assigned for this processor is IBM390PS.
Ready;
```

Response 16 (No Network name):

If there is no network name associated with your processor, you would see:

```
query lpars network
There is no network name assigned to this processor.
Ready;
```

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP1006E Command processing cannot complete because the system does not support the logical-channel-subsystem facility.
- HCP1006E Command processing cannot complete because the system does not support the logical-partition-configuration facility.
- HCP6809E Dynamic I/O configuration is not supported.

# QUERY LSYSTEM

---

► Query — LSYStem ◄

## Authorization

Privilege Class: B

## Purpose

Use QUERY LSYSTEM to display the machine name of the local system. This name is used when identifying two machines connected by CIB (coupling over InfiniBand) channel paths.

## Operands

### LSYStem

requests the display of the machine name of the local system.

## Responses

Response 1:

If you enter QUERY LSYSTEM and the name of your system is CPCSYS43, the response is:

```
q lsys
The name of this system is CPCSYS43.
Ready;
```

Response 2:

If you enter QUERY LSYSTEM and no name has been assigned to your system, the response is:

```
q lsys
This system has no name assigned.
Ready;
```

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP1006E Command processing cannot complete because the system does not support the coupling over InfiniBand channel-path facility.
- HCP6815E Command failed in channel subsystem.

## QUERY MAXLDEV

---

► Query — MAXLdev ◄

### Authorization

Privilege Class: E

### Purpose

Use QUERY MAXLDEV to display the maximum number of logical devices that can exist at the same time.

### Usage Notes

1. The number of logical devices may be greater than the maximum specified on the command response. This can happen when the limit is lowered to a value less than the current number of logical devices. If this occurs, the existing logical devices will be unaffected, but no new logical devices can be created until the number of logical devices is less than the maximum allowed.
2. All logical devices will be created with a device address less than the limit in effect.

### Responses

Response 1:

```
MAXLDEV = nnnnn
```

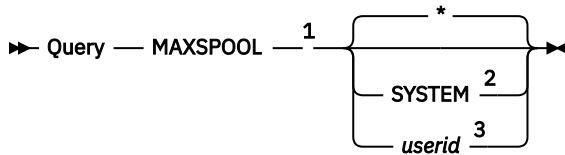
#### ***nnnnn***

identifies the maximum number of logical devices that can be created.

### Messages

- HCP026E Operand missing or invalid

## QUERY MAXSPOOL



### Notes:

- <sup>1</sup> If QUERY MAXSPOOL is executed with the AT command, the SYSTEM or *userid* operand is required.
- <sup>2</sup> SYSTEM is valid only for class D users.
- <sup>3</sup> A class G user may specify only the user's own user ID.

### Authorization

Privilege Class: D, G

### Purpose

Use QUERY MAXSPOOL to display the spool file limit for the system or for any individual user of that system.

### Operands

**\***

displays the maximum number of spool files allowed for the user who issues the command.

### SYSTEM

displays the total number of spool files allowed for the system.

### *userid*

displays the maximum number of spool files allowed for the specified user.

### Usage Notes

1. The user maximum spool file number is set through the SPOOLFILE directory statement.

### Responses

Response 1:

```
MAXIMUM SPOOL FILE COUNT FOR {SYSTEM} IS number
                               {userid}
```

indicates the maximum spool file count allowed.

### SYSTEM

### *userid*

identifies the system or *userid* specified on the QUERY MAXSPOOL command.

### *number*

identifies the maximum spool file count allowed. If you specify SYSTEM, the number is the total number of files allowed in the system. If you specify *userid*, this is the total number of files that this user is allowed to maintain.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid userid - *userid*
- HCP053E [XAUTOLOG failed for *userid*:] *userid/value* not in CP directory
- HCP475I Fatal I/O error trying to read directory from *volid* [for user *userid*]

# QUERY MAXUSERS

---

► Query — MAXUsers ◄

## Authorization

Privilege Class: A

## Purpose

Use QUERY MAXUSERS to display the maximum number of logged-on users allowed.

## Usage Notes

1. The number of users actually logged on may be greater than the maximum value displayed. This can happen when the user logon limit is lowered to a point below the current number of users logged on, when the operator logs on and the maximum number of users is already logged on, or if a virtual machine is logged on that has the IGNMAXU operand of the OPTION control statement included in its user directory.

## Responses

Response 1:

```
MAXUSERS = nnnnn
```

is the response when a user logon limit is in effect.

### **nnnnn**

identifies the maximum number of users allowed to be logged onto the system at any one time.

Response 2:

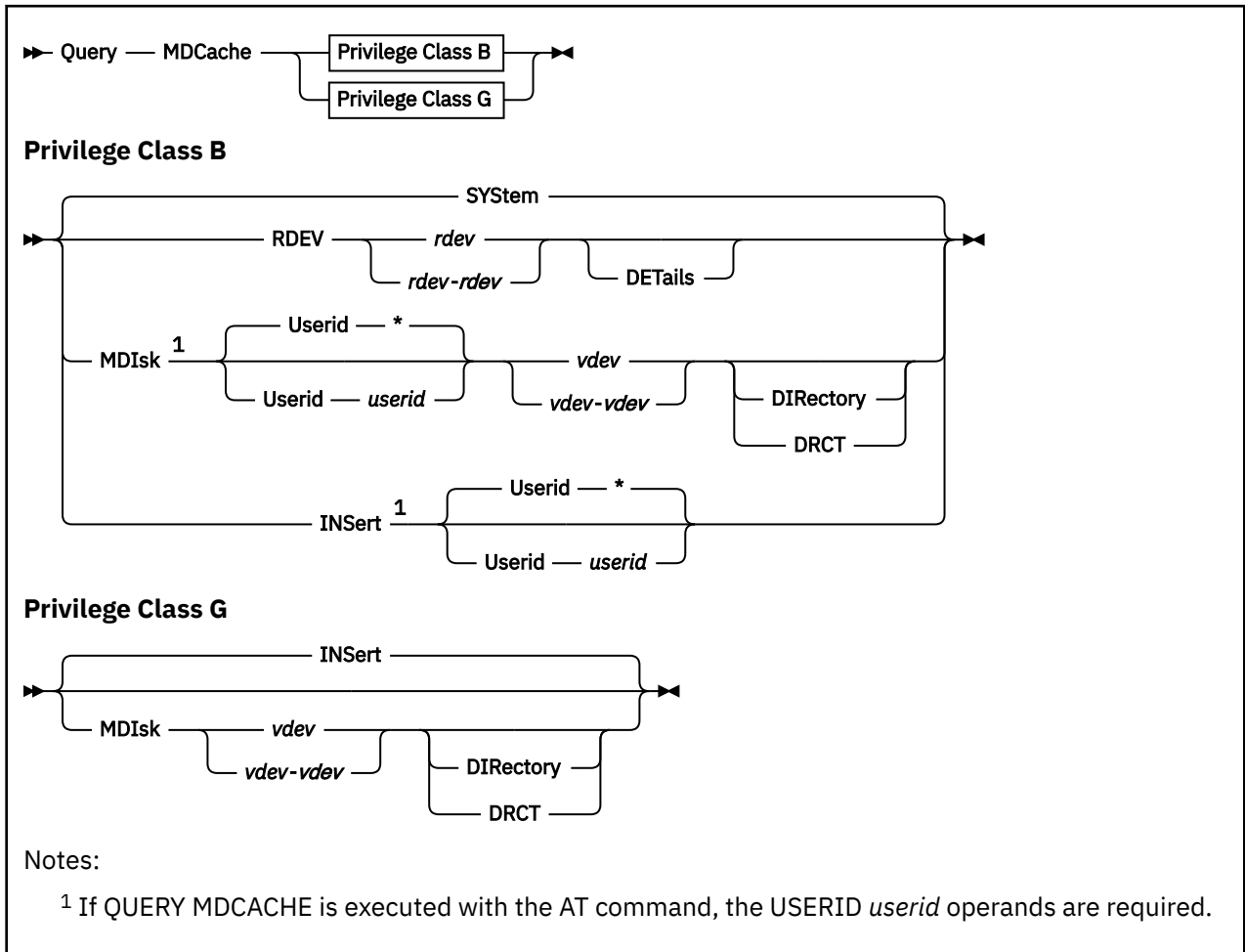
```
NO MAXUSERS LIMIT IS SET
```

is the response if no user logon limit has been set. In that case, the default system maximum of 99,999 users is in effect.

## Messages

- HCP002E Invalid operand - *operand*

## QUERY MDCACHE



### Authorization

Privilege Class: B, G

### Purpose

Use QUERY MDCACHE to display minidisk cache settings or a user's ability to insert data into the cache.

A class B user can query:

- Minidisk cache settings for the entire system, a real device, an active minidisk, or a minidisk defined in the directory.
- Another user's ability to insert data into the cache.

A class G user can query:

- The user's ability to insert data into the cache.
- An active minidisk or a minidisk defined in the user's directory entry.

### Operands

#### SYSTEM

indicates that the cache setting for the system is being queried.



**RDEV**

indicates that the cache settings for one or more real devices are being queried.

**rdev****rdev-rdev**

identifies the real device numbers whose cache eligibility is being queried.

**DEtails**

indicates that cache settings for any active minidisks on the real device(s) being queried will be displayed in the response.

**MDisk**

indicates that the cache settings for active minidisks are being queried.

**Userid \*****Userid *userid***

is the user ID of the virtual machine for which you want to query minidisk cache settings. If omitted or specified as an asterisk (\*), CP uses the user ID of the virtual machine issuing the command. For the class B command, if DRCT or DIRectory is not specified, the userid specified must be logged on.

**vdev****vdev-vdev**

identifies the virtual device number or the range of virtual device numbers for which cache settings are to be queried. For the class B command, the virtual device numbers specified correspond to the userid's current configuration if the DIRECTORY or DRCT option is not specified and to the userid's directory entry if the DIRECTORY or DRCT option is specified. For the class G command, the virtual device numbers specified correspond to the userid's directory entry.

**DIRectory****DRCT**

indicates that the virtual device numbers correspond to the userid's directory entry. For the class B command, if DRCT or DIRectory is not specified, the userid specified must be logged on.

**INSert**

queries a user's ability to insert information into the cache.

**Usage Notes**

1. In z/Architecture mode, minidisk cache uses host real storage above the 2 GB line.

**Responses**

Response 1:

The following response is issued when you specify Query MDCache from a userid with privilege class B and caching is enabled for the system.

```
Minidisk cache ON for system
Storage MDC min=nnnM max=nnnM, usage=ppp%, bias=b.bb
```

**min=nnnM**

indicates the current minimum value for the size of minidisk cache in megabytes.

**max=nnnM**

indicates the current maximum value for the size of minidisk cache in megabytes.

**usage=ppp%**

indicates the percentage of real storage in use by minidisk cache.

**bias=bb.b**

indicates the bias factor used by the arbiter to determine the target size for the storage type.

Response 2:

The following response is issued when you specify Query MDCache from a userid with only privilege class G.

```
Minidisk cache inserts {ON } for userid
                       {OFF }
```

***userid***

identifies the *userid* where the command was issued.

Response 3:

The following response is issued when you specify Query MDCache SYStem and caching is disabled for the system.

```
Minidisk cache OFF for system
```

Response 4:

The following response is issued when you specify Query MDCache RDEV *rdev*.

```
Minidisk cache {DFLTON } for rdev
                {DFLTOFF }
                {OFF   }
```

***rdev***

identifies the real device number of the device being queried.

Response 5:

The following response is issued when you specify Query MDCache RDEV *rdev-rdev*.

```
Minidisk cache {DFLTON } for rdev-rdev
                {DFLTOFF }
                {OFF   }
```

***rdev-rdev***

identifies the first and last real device numbers in the range.

Response 6:

The following response is issued for each active real device when you issue Query MDCache RDEV *rdev* DETails or Query MDCache RDEV *rdev-rdev* DETails. The additional lines for each real device contain cache eligibility information for any active minidisks on that device. The first ON/OFF for each minidisk indicates whether caching is currently enabled for the minidisk. The information in parenthesis which indicates that cache setting for the minidisk plus the cache setting for the real device add up the current cache eligibility of the minidisk.

```
Minidisk cache {DFLTON } for rdev {ON } ({ON      } for userid vdev)
                {DFLTOFF }          {OFF }  {OFF     }
                {OFF   }            {OFF }  {DEFAULT }
```

***rdev***

identifies the real device number of the device being queried.

***userid***

identifies the owner of an active minidisk on the device being queried.

***vdev***

identifies the owning device number of an active minidisk on the device being queried.

Response 7:

The following response is issued for each device when you specify Query MDCache MDIsk *vdev* or Query MDCache MDIsk *vdev-vdev*. The first ON/OFF for each minidisk indicates whether caching is currently enabled for the minidisk. The information in parenthesis indicates the cache setting for the minidisk, provides minidisk owner information, and indicates the cache setting for the real device. The minidisk and real device settings combine to indicate the current cache eligibility of the minidisk. In an SSI cluster,

you will receive additional information in parenthesis when MDC is suppressed due to a write link from another system in the SSI cluster.

```
DASD vdev MDCACHE {ON } ({ON } for userid ovdev; {DFLTON }
                {OFF} {OFF} {DEFAULT} {DFLTOFF}
Continued:
for DASD rdev) [recordmdc enabled] (MDC suppressed due to WRITE link
on {system})
```

***ivdev***

is the *vdev* from the command line whose minidisk cache status is provided. When the DIRECTORY option is specified, *ivdev* is with respect to the directory. When DIRECTORY is not specified, *ivdev* is with respect to the user configuration specified or implied on the command line.

***userid***

identifies either the user whose directory entry has the MDISK statement for this minidisk or the user who issued the DEFINE command that defined the minidisk. This is the *userid* that 'owns' the minidisk.

***ovdev***

identifies the device number from either the MDISK statement or as specified on the DEFINE command. This is the *vdev* from the perspective of the owning *userid*.

***rdev***

identifies the real device on which the virtual device resides.

**recordmdc enabled**

indicates that record level caching rather than normal full track caching is active for this minidisk.

Response 8:

The following response is issued when you specify Query MDCache INSert Userid *userid* or the class G command Query MDCache INSert.

```
Minidisk cache inserts {ON } for userid
                    {OFF}
```

***userid***

identifies the *userid* whose ability to insert into the cache is being queried.

Response 9:

The following response is issued when you specify Query MDCache MDIsk *vdev1-vdev2* and caching was last disabled by Transport Mode I/O.

```
DASD vdev1 MDCACHE ON (DEFAULT for userid vdev1; DFLTON for DASD rdev1)
DASD vdev2 MDCACHE OFF (OFF for userid vdev2; DFLTON for DASD rdev2)
Disabled by Transport Mode I/O
```

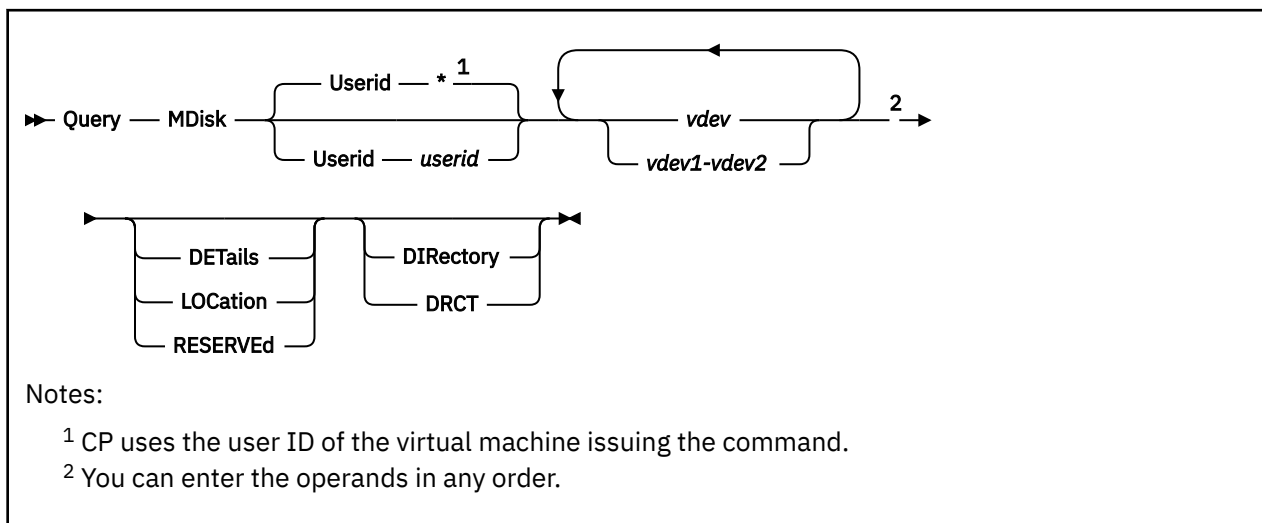
***userid***

identifies the *userid* whose ability to insert into the cache is being queried.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP0013E Conflicting option - *option*
- HCP0026E Operand missing or invalid
- HCP0045E *userid* not logged on
- HCP1920E Minidisk cache not enabled for system
- HCP1921E Cache settings could not be {CHANGED|QUERIED} for {*rdev* code. *cc*|*userid* *vdev* code *cc*|*rdev- rdev*|*userid* *vdev-vdev*}

## QUERY MDISK



### Authorization

Privilege Class: G

### Purpose

Use QUERY MDISK to display the device characteristics of minidisks: dedicated DASD, full-pack minidisks, user (permanent) minidisks, temporary disks (T-disks), and virtual disks in storage.

### Operands

#### Userid \*

#### Userid *userid*

is the user ID of the virtual machine for which you want to display minidisk information. If omitted or specified as an asterisk (\*), CP uses the user ID of the virtual machine issuing the command.

#### *vdev*

#### *vdev1-vdev2*

specifies the virtual device number (or numbers) of the minidisks for which you want to see information.

#### DEtails

tells CP you want information about the type of minidisk and the duplex control operations performed for it. If you specify DETAILS, you cannot specify LOCATION.

#### LOCation

tells CP you want information about the location and size of the specified minidisk. If you specify LOCATION, you cannot specify DETAILS.

#### RESERVEd

tells CP you want information about who holds a reserve on a minidisk.

#### DIRectory

#### DRCT

tells CP you want information about the minidisk defined in the CP user directory, rather than information about the currently defined minidisks for the target user ID. The minidisk must be on a dasd that is online and attached to the system.

## Usage Notes

1. For more information about the DEVNO information that CP displays when you specify DETAILS, see the [MDISK Statement in z/VM: CP Planning and Administration](#).
2. For more information on the duplex control operations that CP provides when you specify DETAILS, see “DUPLEX” on page 448.
3. The OwnerID and Odev listed in the output can be different, depending on how the disk was defined. If the disk definition in the system directory has been changed, the new OwnerID and Odev may not be immediately reflected in the QUERY MDISK output. The old designation will be shown until there are no users accessing the disk. Use the form of the command consistent with the output needed for your application.

## Responses

Response 1:

If you do not specify DETAILS or LOCATION, you see this type of response:

```
query mdisk 190
TargetID Tdev OwnerID Odev
RICH      0190 MAINT    0190
Ready;
```

This example tells you that RICH's 190 disk is really MAINT's 190 disk (MAINT is the owner of the disk).

Response 2:

If you specify DETAILS, CP adds more information to the response:

```
query mdisk 190 details
TargetID Tdev OwnerID Odev Minidisk DEVNO Duplex
RICH      0190 MAINT    0190 Regular   No    No
Ready;
```

This example tells you MAINT's 190 disk is a regular minidisk, it does not have the DEVNO operand of the MDISK statement defined in the user's directory entry, and no duplex control operations are in effect for this disk.

The possible values for the Minidisk column are:

### Dedicated

The real device associated with the specified virtual device is dedicated.

### Full pack

The specified virtual device is a full-pack minidisk.

### Regular

The specified virtual device is a non-full-pack minidisk. **Note:** This category includes T-disks and virtual disks in storage.

The possible values for the DEVNO column are:

### No

The minidisk is not defined using the DEVNO operand of the MDISK statement in the user's directory entry.

### Yes

The minidisk is defined using the DEVNO operand of the MDISK statement in the user's directory entry.

The possible values for the Duplex column are:

### No

The minidisk has no duplex control operations in effect.

## Primary

The minidisk is the primary device of a duplex pair.

## Secondary

The minidisk is the secondary device of a duplex pair.

Response 3:

If you specify LOCATION, CP displays information about the volume on which the minidisk resides. For example:

```
query mdisk 190 location
TargetID Tdev OwnerID Odev Dtype Vol-ID Rdev StartLoc Size
RICH      0190 MAINT    0190 3380 XAUSR5 0284      2276    40
Ready;
```

gives you information about the:

- Device type of the real disk on which the minidisk resides; for a virtual disk in storage, this is the device type of the device simulated in storage.
- Volume label of the real disk; for a virtual disk in storage, this is always (VDSK).
- Real device address of the disk; for a virtual disk in storage, this is always VDSK.
- Starting offset of the minidisk (in cylinders or blocks) into the real disk; for a virtual disk in storage, this is always 0.
- Size (in cylinders or blocks) of the minidisk.

Response 4:

In an SSI cluster, if you specify DETAILS, CP adds more information to the response:

```
query mdisk 190 details
TargetID Tdev OwnerID Odev Minidisk DEVNO Duplex Scope
REESELIS 0191 REESELIS 0191 Regular No No Local
REESELIS 0495 REESELIS 0495 Regular No No Global
REESELIS 0690 REESELIS 0690 Regular No No Local - DEFINE MDISK
Ready;
```

The possible values for the Scope column are:

### Local

The minidisk is restricted to a particular system in the SSI cluster.

### Global

The minidisk is not restricted to a particular system in the SSI cluster.

### Local - DEFINE MDISK

The DEFINE MDISK command has been used to create the minidisk.

Response 5:

The RESERVED operand will display one of the following based on the reserved state of each specified minidisk:

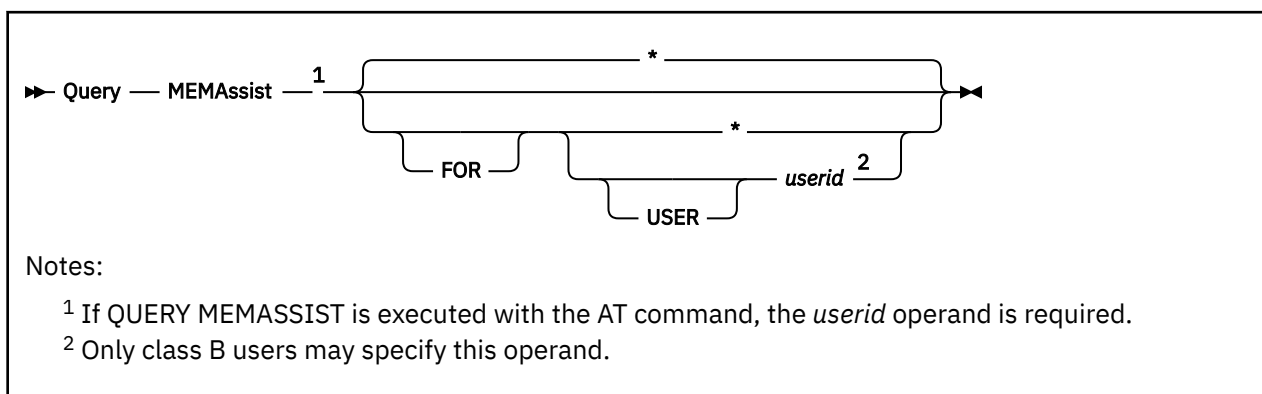
```
NO RESERVE HELD FOR vdev
vdev RESERVED BY userid
```

## Messages

- HCP003E Invalid option - *option*
- HCP007E Invalid userid - *userid*
- HCP009E Invalid range - *range*
- HCP009E No disks found among the input virtual device numbers
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid

- HCP022E A virtual device number was not supplied or it was invalid
- HCP040E Device *address* does not exist
- HCP045E *userid* not logged on
- HCP052E Error in CP directory
- HCP053E *userid* not in CP directory
- HCP410E Command terminated; paging error
- HCP1008E Device *address* is not a DASD
- HCP1160E Device *vdev* has excessive indirect links in CP directory.

## QUERY MEMASSIST



### Authorization

Privilege Class: B, G

### Purpose

Use QUERY MEMASSIST to determine the collaborative memory management assist settings for your own virtual machine or another user's virtual machine.

### Operands

\*

indicates that you wish to display your own virtual machine's MEMASSIST settings.

#### **USER** *userid*

is the user ID of the virtual machine for which you want to display the MEMASSIST settings.

### Usage Notes

1. The initial MEMASSIST settings for your virtual machine are determined as follows:
  - a. If the collaborative memory management assist is installed on the system, the initial setting is ON unless the OPTION NOMEMASSIST is specified in the user directory entry.
  - b. If the collaborative memory management assist is not installed, the initial setting is OFF.
2. At z/VM IPL, the initial MEMASSIST FOR ALL setting is ON, even if the assist is not installed on the machine. This allows a virtual machine to SET MEMASSIST ON, causing the assist to be simulated by z/VM for guest testing purposes.
3. You can use the class G SET MEMASSIST command to change your MEMASSIST setting. You can use the class B SET MEMASSIST command to change another virtual machine's MEMASSIST setting. For more information, see [“SET MEMASSIST” on page 1568](#).
4. If the collaborative memory management assist is not installed on the machine, guests can still use it by means of the simulation that is built into CP. However, because of the overhead incurred by simulation, this mode of operation is recommended for testing purposes only.

### Responses

The response displays the overall setting for all users, followed by the setting and status for the individual user. The response pattern is:

```
ALL USERS SET - {ON | OFF}
```



USER	SETTING	STATUS
<i>userid</i>	<i>setting</i>	<i>status</i>

The possible *setting* and *status* combinations are:

SETTING	STATUS
OFF	INACTIVE
ON	INACTIVE
ON	INACTIVE,PENDING OFF
(ON)	INACTIVE,PENDING OFF
ON	ACTIVE
ON	ACTIVE,PENDING OFF
(ON)	ACTIVE,PENDING OFF

Any of these may be suffixed by *, SIMULATED*.

**ALL USERS SET - {ON | OFF}**

The setting for ALL USERS is either ON or OFF. The setting is controlled by the class B command SET MEMASSIST ON|OFF FOR ALL, which controls access to the collaborative memory management assist on a system-wide basis.

ON means users can enable the assist for their virtual machines through the SET MEMASSIST ON command and use the assist by executing the EXTRACT AND SET STORAGE ATTRIBUTES (ESSA) instruction.

OFF means users cannot enable the assist for their virtual machines (SET MEMASSIST ON is rejected), and users who are not currently using the assist are not able to begin using it (the ESSA instruction causes an operation exception). However, users who are currently using the assist continue to have access to it until the next virtual machine reset.

**USER**

The *userid* field contains the user ID of the virtual machine being queried.

**SETTING**

This field is ON, (ON), or OFF. This setting is independent of the system-wide setting. That is, the system-wide setting is like one gate to pass through, which can be open or closed. The user setting is like a second gate, which can be open or closed independently of the first gate.

ON means the user has access to the assist and may begin using it if not already doing so.

(ON) means that the user's current setting is ON, but system-wide access to the assist is denied because that setting is OFF. The user may continue using the assist if already doing so.

OFF means the user does not have access to the assist and that execution of the ESSA instruction results in an operation exception.

The default value for this field is determined during logon processing and is set ON if the assist is installed on the machine and OPTION NOMEMASSIST does not appear in the user's directory entry; it is set OFF otherwise.

**STATUS**

The status field indicates the current disposition of the virtual machine with respect to the assist. The field can be INACTIVE; INACTIVE,PENDING OFF; ACTIVE; or ACTIVE,PENDING OFF. In addition, SIMULATED is appended to the status, if applicable.

INACTIVE means the user is not currently using the assist.

ACTIVE means the user is currently using the assist (that is, the guest has executed at least one ESSA instruction since the last virtual machine reset).

PENDING OFF means the user might be currently using the assist, but that the SET MEMASSIST OFF command has been issued by or for the user. The user might begin or continue using the assist until the next virtual machine reset, at which time the SET MEMASSIST OFF command takes effect.

SIMULATED means the assist is not installed on the machine, and that it is simulated by CP.

Sample responses are:

## QUERY MEMASSIST

**Response 1:** In the following response, the assist is SET ON FOR ALL, SET ON FOR LINUX01, and the software in the LINUX01 virtual machine has begun using the assist.

```
ALL USERS SET - ON
USER      SETTING  STATUS
LINUX01  ON           ACTIVE
```

**Response 2:** In the following response, the assist is SET ON FOR ALL and SET ON FOR MANSELL, but the software in the MANSELL virtual machine is not using the assist.

```
ALL USERS SET - ON
USER      SETTING  STATUS
MANSELL  ON           INACTIVE
```

**Response 3:** In the following response, the assist is SET OFF FOR ALL, superseding the current setting of ON FOR MANSELL. The virtual machine is not now using the assist, and is not permitted to start using it (the assist appears not available to the virtual machine).

```
ALL USERS SET - OFF
USER      SETTING  STATUS
MANSELL  (ON)       INACTIVE, PENDING OFF
```

**Response 4:** In the following response, the assist is SET OFF FOR ALL, which would normally supersede the individual user setting, ON FOR MANSELL. Because the virtual machine had already begun using the assist before it was SET OFF FOR ALL, it is permitted to continue using it until the next virtual machine reset. At that point, the assist will become INACTIVE, and will appear not available to the virtual machine.

```
ALL USERS SET - OFF
USER      SETTING  STATUS
B         (ON)       ACTIVE, PENDING OFF
```

### Messages

- HCP003E Invalid option - *option*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on

# QUERY MITIME

► Query — MITime ◄

## Authorization

Privilege Class: A, B

## Purpose

Use QUERY MITIME to determine the current time interval settings for the missing interrupt detector.

## Operands

### MITime

requests the display of the current time interval settings for the missing interrupt detector.

## Usage Notes

1. When you enter QUERY MITIME, each current MITIME setting is displayed in a form similar to the one you used when you issued the SET MITIME command to invoke the specifications.
2. When multiple responses are displayed for specific devices because of the overlap in their range and class, the first response takes precedence, because the command displays previous SET MITIME settings in a last-in, first-out sequence.
3. For storage subsystems that provide a primary time-out value for missing interrupt handler, a response is displayed for each device or range of devices defined in the same storage subsystem. It is recommended that all devices belonging to the same storage subsystem be defined in a single ascending range.

## Responses

Response 1:

*Example of Multiple Responses:* Assume the valid range of DASD addresses is 190 through 400 and several SET MITIME commands were entered previously. If you wish to determine the current missing interrupt time interval for DASD 251 and 291, enter `query mitime`. If the response is as follows:

```
MITIME ON 250-252 00:10
MITIME ON 190-255 01:40
MITIME ON DASD 00:30,
```

the current time interval for device 251 is 10 seconds, and the current time interval for device 291 is 30 seconds.

```
MITIME {ON } {devclass } [mm:ss]
      {OFF} {rdev }
          {rdev-rdev}
          {ALL }
```

### ON

### OFF

indicates whether time intervals for the specified devices have been set on. ON specifies that the intervals have been set; OFF specifies that time intervals have been set off and that no monitoring for missing interrupts is being done for the specified devices.

## QUERY MITIME

### ***devclass***

indicates the device class for which you want to display the missing interrupt detector settings. The operand *devclass* may be DASD, TAPE, GRAF, DISPLAY, UR, SWITCH, or MISC. GRAF and DISPLAY do not include the 328x, and UR does not include the 3800. MISC includes 328x, 3800, and any DASD defined as a mass storage subsystem device.

### ***rdev***

#### ***rdev-rdev***

is a real device number or range or real device numbers for which you want to display the missing interrupt detector settings.

### **ALL**

indicates that the missing interrupt detector setting is in effect for all of your devices.

### ***mm:ss***

indicates the time interval in minutes and seconds. The maximum value for *mm* is 99, while the maximum value for *ss* is 59. This operand is displayed only if MITIME is set ON.

## **Messages**

- HCP002E Invalid operand - *operand*

## QUERY MONDATA

---

► Query — MONData ◄

### Authorization

Privilege Class: C

### Purpose

Use QUERY MONDATA to determine if user display data would be present in monitor event records for terminals.

### Usage Notes

1. To change the MONDATA option, enter the SET MONDATA command.

### Responses

Response 1:

```
MONDATA {ON }  
         {OFF}
```

#### **ON**

indicates that input from, and output to, user display stations is included with the terminal event records. It is the responsibility of the installation to protect the security of this data.

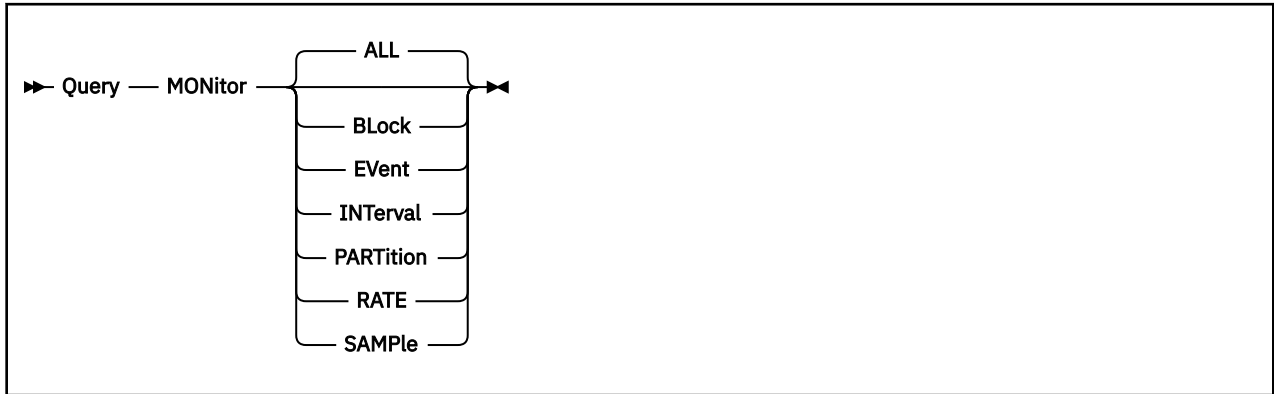
#### **OFF**

indicates that input from, and output to, user display stations is not included with the terminal event records.

### Messages

- HCP002E Invalid operand - *operand*

## QUERY MONITOR



### Authorization

Privilege Class: A, E

### Purpose

Use QUERY MONITOR to display the current profile information for monitor.

### Operands

#### ALL

requests a display of all settings and information about CP monitoring. This includes both event and sample monitoring. This is the default.

#### BLock

requests a display of the current number of 4 KB pages of the saved segment that are accumulated before an IUCV message is sent to the virtual machine indicating that there is event data to be read.

The value displayed is as set or defaulted by the BLOCK option of the MONITOR command.

#### EVent

requests a display of all current settings established for event monitoring. Included in this display are the settings, listings, and status of applicable domains and their elements, users connected to \*MONITOR for event data collections, and partitioning of the saved segment, among other things.

#### INTerval

requests a display of the interval value currently in effect for sample monitoring. This value is as set or defaulted by the INTERVAL option of the MONITOR SAMPLE command.

#### PARTition

requests a display of the size of a saved segment that is currently partitioned for event records. The value is as set or defaulted by the PARTITION option of the MONITOR EVENT or MONITOR START command.

#### RATE

requests a display of the rate value currently in effect for high frequency sample monitoring. This value is as set or defaulted by the RATE option of the MONITOR SAMPLE command.

#### SAMPlE

requests a display of all current settings established for sample monitoring. Included in this display are the settings, listings, and status of applicable domains and their elements, users connected to \*MONITOR for sample data collections, and partitioning of the saved segment, among other things.

**Examples**

**Note:** In these examples, leading zeros are converted to blanks, and blank spaces and indents have been added to improve readability.

Example for QUERY MONITOR SAMPLE:

```

MONITOR SAMPLE ACTIVE      NEXT 12/15/2022 11:00:00
                        INTERVAL 30 SECONDS  PENDING INTERVAL 20 SECONDS
                        RATE 11.21 SECONDS
MONITOR DCSS NAME - MONDCSS
CONFIGURATION SIZE 50    LIMIT 2 MINUTES
CONFIGURATION AREA BE FREED WITHIN - 1 MINUTES 30 SECONDS
USERS CONNECTED TO *MONITOR - USER45
                                USER46 QUIESCED

MONITOR   DOMAIN ENABLED
SYSTEM   DOMAIN ENABLED
PROCESSOR DOMAIN DISABLED
STORAGE  DOMAIN DISABLED
USER     DOMAIN ENABLED
    THE FOLLOWING USERIDS ARE ENABLED:
        ABLEFOX MAINT USER01  USER02  USER03  USER04  USER05
        WILLIAMS WILSON
I/O      DOMAIN ENABLED
    PCIF CLASS ENABLED
    THE FOLLOWING DEVICES ARE ENABLED:
        01A0-01AF 08C0-08FF 0900 0934-093F
        0999-1010 1030-104F
        2222      3000-32FF 3400-3404
NETWORK  DOMAIN DISABLED
ISFC     DOMAIN DISABLED
APPLDATA DOMAIN ENABLED
    THE FOLLOWING USERIDS ARE ENABLED:
        APPLID  USER06  USER07
SSI       DOMAIN DISABLED
    
```

**SAMPLE**

describes the monitor sample domain.

**ACTIVE**

indicates that sample recording is active.

**NEXT 12/15/2022 11:00:00**

indicates that the next sample collection will begin on date 12/15/2022 at time 11:00:00.

**INTERVAL 30 SECONDS**

indicates that the current monitor sampling interval is set to 30 seconds.

**PENDING INTERVAL 20 SECONDS**

indicates that a new sampling interval of 20 seconds takes effect after the current sampling interval is completed.

**RATE 11.21 SECONDS**

indicates that high-frequency sampling is active, with a current rate of 11.21 seconds. The *first* leading zero in the RATE value is converted to a blank, if applicable (for example, 8.21 not 08.21).

**MONITOR DCSS NAME - MONDCSS**

indicates that the name of the DCSS used to collect monitor data is MONDCSS.

**CONFIGURATION SIZE 50 LIMIT 2 MINUTES**

indicates 50 pages are allocated for sample configuration records and that users have two minutes to reply to the sample configuration IUCV message.

**CONFIGURATION AREA WILL BE FREED WITHIN - 1 MINUTE 30 SECONDS**

indicates that the configuration area is freed for any pending users within 1 minute and 30 seconds. Users processing configuration records must reply within this time, or the IUCV configuration message is purged, and the associated pages are released for reuse.

**USERS CONNECTED TO \*MONITOR - USER45 ... USER46 QUIESCED**

indicates that users USER45, USER46, USER47, and USER48 are connected to \*MONITOR to receive sample data notifications.

## QUERY MONITOR

- USER45's IUCV path to \*MONITOR is not quiesced. This user is eligible to receive sample data notifications.
- USER46's IUCV path to \*MONITOR is quiesced. This user is eligible to receive sample data notifications.
- USER47's IUCV path to \*MONITOR is not quiesced. This user has not replied to the IUCV send for the configuration records. USER47 is eligible to receive data.
- USER48's IUCV path to \*MONITOR is not quiesced. This user is waiting for the sample configuration area to free up. (The area is freed when USER47 replies or the configuration time limit is reached.) USER48 is not eligible to receive data notification, but becomes eligible after having been notified of configuration records.

Since there is more than one user connected, you know \*MONITOR is executing in shared mode. The user IDs are displayed in alphabetic order.

### **MONITOR DOMAIN ENABLED**

indicates that the monitor domain is enabled.

### **SYSTEM DOMAIN ENABLED**

indicates that the system domain is enabled.

### **PROCESSOR DOMAIN DISABLED**

indicates that the processor domain is disabled.

### **STORAGE DOMAIN DISABLED**

indicates that the storage domain is disabled.

### **USER DOMAIN ENABLED**

indicates that the user domain is enabled.

### **ABLEFOX MAINT USER01 USER02 USER03 USER04 USER05 WILLIAMS WILSON**

indicates that there are nine users currently being monitored: ABLEFOX, MAINT, USER01, USER02, USER03, USER04, USER05, WILLIAMS, and WILSON. The user IDs are displayed in alphabetic order.

### **I/O DOMAIN ENABLED**

indicates that the I/O domain is enabled.

### **PCIF CLASS ENABLED**

indicates that the PCIF class is enabled.

### **01A0-01AF 08C0-08FF 0900 0934-093F 0999-1010 1030-104F 2222 3000-32FF 3400-3404**

indicates that device 0900 and device 2222 are being monitored. Also, all devices in the following ranges are being monitored: 01A0-01AF, 08C0-08FF, 0934-093F, 0999-1010, 1030-104F, 3000-32FF, 3400-3404.

Devices are shown in ascending order. All leading zeros, if applicable, are shown. For example, device number 3 is displayed as 0003.

### **NETWORK DOMAIN DISABLED**

indicates that the NETWORK domain is disabled.

### **ISFC DOMAIN DISABLED**

indicates that the ISFC domain is disabled.

### **APPLDATA DOMAIN ENABLED**

indicates that the APPLDATA domain is enabled.

### **APPLID USER06 USER07**

indicates that there are three users whose application data is being collected, if a DIAGNOSE X'DC' START has been issued: APPLID, USER06, and USER07.

### **SSI DOMAIN DISABLED**

indicates that the SSI domain is disabled.



## Responses

**Note:** To make it easier to find the various responses, every response is listed by option (ALL, BLOCK, EVENT, INTERVAL, PARTITION, RATE, and SAMPLE).

Response 1 (ALL):

When QUERY MONITOR ALL is issued, the following responses are displayed:

- Response 3 (EVENT)
- Response 7 (SAMPLE)

Response 2 (BLOCK):

If QUERY MONITOR BLOCK is specified, the response is:

```
MONITOR BLOCK n
```

### BLOCK n

shows the maximum number of 4 KB pages accumulated before an IUCV message is sent to the virtual machine indicating that there is event data to be read.

Response 3 (EVENT):

If QUERY MONITOR EVENT is specified, the response is:

```
MONITOR EVENT {ACTIVE    } BLOCK n PARTITION n
               {INACTIVE  }
               {SUSPENDED}
               {PENDING   }

MONITOR DCSS NAME - {dcssname          }
                   {NO DCSS NAME DEFINED}

CONFIGURATION  SIZE n          LIMIT {n MINUTES}
                                       {n SECONDS}
CONFIGURATION AREA {WILL BE FREED WITHIN - {n MINUTES n SECONDS}}
                  {IS FREE}

USERS CONNECTED TO *MONITOR {NO USERS CONNECTED}
                            {userid[QUIESCED] [PROCESSING-CONFIG] [EXCLUSIVE]}
                            [PENDING-CONFIG ]
```

### EVENT

describes the status of event monitoring.

#### ACTIVE

indicates that EVENT recording is active.

#### INACTIVE

indicates that EVENT monitoring is not active.

#### SUSPENDED

indicates that EVENT monitoring is suspended either because monitor is in exclusive mode and the user connected has quiesced the path to \*MONITOR and/or the event partition of the DCSS is full; or because there are no CP frames available to write event data.

#### PENDING

indicates that EVENT recording is not being performed because either there is no eligible monitor user to receive data notification for event data or sample data is still in the event area of the DCSS.

#### BLOCK n

indicates the maximum number of 4 KB pages accumulated before an IUCV message is sent to the virtual machine, indicating there is event data to be read.

#### PARTITION n

indicates the number of 4 KB pages allocated for the entire event area (configuration and data record pages). If no users are connected, and event monitoring has been started with no partition specified, n is displayed as 50% to show that the default size of 50% of the DCSS size is used.

**MONITOR DCSS NAME**

indicates the name of DCSS to collect monitor data.

***dcssname***

is the name of the DCSS being used.

**NO DCSS NAME DEFINED**

indicates that no DCSS name is defined.

**CONFIGURATION SIZE *nn***

indicates the number of 4 KB pages (within the event partition) allocated for event configuration records. Leading zeros, if any, are removed.

**LIMIT *n* MINUTES**

**LIMIT *n* SECONDS**

displays the time limit in which a reply is required to the configuration IUCV message. If the value does not fall on a whole minute, the value is displayed in seconds, and the suffix SECONDS is displayed. Leading zeros, if any, are removed.

**CONFIGURATION AREA WILL BE FREED WITHIN - *n* MINUTES *n* SECONDS**

displays the amount of time that the users holding the configuration area have left to reply to the configuration IUCV message.

**CONFIGURATION AREA IS FREE**

is displayed if no users are holding the configuration area.

**USERS CONNECTED TO \*MONITOR**

lists, in alphabetic order, the user IDs of the users connected to \*MONITOR to collect event data.

**NO USERS CONNECTED**

indicates that no user is connected to \*MONITOR to collect event data.

***userid***

identifies the users connected to \*MONITOR for event data.

**QUIESCED**

indicates that the user is not allowing \*MONITOR to send IUCV messages over this path to the user's virtual machine.

**Note:** If a user's path is at its IUCV message limit, the user remains quiesced) until the user responds to previous messages from \*MONITOR.

**EXCLUSIVE**

indicates that monitor is running in exclusive mode, not shared mode. The mode is specified by the format of the connect interface used by the first user that connects to \*MONITOR.

The following response lines are given based on the specifications in the event profile:

```
MONITOR   DOMAIN  ENABLED
PROCESSOR DOMAIN  {DISABLED}
           {ENABLED }
STORAGE   DOMAIN  {DISABLED}
           {ENABLED }
SCHEDULER DOMAIN  {DISABLED}
           {ENABLED }
```

**Note:** The following additional responses are issued only if the SCHEDULER domain is enabled, based on the specifications for the SCHEDULER domain in the event profile:

```
ALL USERS  ENABLED
```

- or -

```
ALL USERS EXCEPT:
  userid [userid...]
```

- or -

THE FOLLOWING USERIDS ARE ENABLED:  
*userid [userid...]*

User IDs are displayed in alphabetic order.

SEEKS      DOMAIN {DISABLED}  
                                   {ENABLED }

**Note:** The following additional responses are issued only if the SEEKS domain is enabled, based on the specifications for the SEEKS domain in the event profile:

ALL DEVICES ENABLED

- or -

ALL DEVICES EXCEPT:  
*rdev[-rdev] or [rdev... rdev]*

- or -

THE FOLLOWING DEVICES ARE ENABLED:  
*rdev[-rdev]*

USER      DOMAIN {DISABLED}  
                                   {ENABLED }

**Note:** The following additional responses are issued only if the USER domain is enabled, based on the specifications for the USER domain in the event profile:

ALL USERS ENABLED

- or -

ALL USERS EXCEPT:  
*userid [userid...]*

- or -

THE FOLLOWING USERIDS ARE ENABLED:  
*userid [userid...]*

I/O      DOMAIN {DISABLED}  
                                   {ENABLED }

**Note:** The following additional responses are issued only if the I/O domain is enabled, based on the specifications for the I/O domain in the event profile:

PCIF CLASS ENABLED

- or -

PCIF CLASS DISABLED

- and -

ALL DEVICES ENABLED

- or -

ALL DEVICES EXCEPT:  
*rdev[-rdev] or [rdev... rdev]*

- or -

THE FOLLOWING DEVICES ARE ENABLED:  
*rdev[-rdev]*

**Note:**

1. For the APPLDATA domain, a user is enabled only if the OPTION APPLMON statement is specified in its directory entry.
2. The following additional responses are issued only if the APPLDATA domain is enabled, based on the specifications for the APPLDATA domain in the event profile:

```
ALL USERS ENABLED
```

- or -

```
ALL USERS EXCEPT:
  userid [userid...]
```

- or -

```
THE FOLLOWING USERIDS ARE ENABLED:
  userid [userid...]
```

```
APPLDATA  DOMAIN  {DISABLED}
             {ENABLED }
```

```
NETWORK    DOMAIN  {DISABLED}
             {ENABLED }
```

```
ISFC       DOMAIN  (DISABLES}
             {ENABLED }
```

```
SSI        DOMAIN  {DISABLED}
             {ENABLED }
```

```
COMMAND    DOMAIN  {DISABLED}
             {ENABLED }
```

Response 4 (INTERVAL):

If QUERY MONITOR INTERVAL is specified, the response is:

```
MONITOR INTERVAL {n MINUTES} PENDING INTERVAL [n MINUTES]
                 {n SECONDS}  [n SECONDS]
```

**INTERVAL *n* MINUTES**

**INTERVAL *n* SECONDS**

describes the current monitor sampling rate, where *n* is the current interval value (1 through 60 for minutes, 6 through 3600 for seconds). If an INTERVAL value does not fall on a whole minute, the value is displayed in seconds, and the suffix SECONDS is displayed.

**PENDING INTERVAL *n* MINUTES**

**PENDING INTERVAL *n* SECONDS**

indicates that a new sampling interval is to take effect after the current sampling interval has occurred. If the new INTERVAL value does not fall on a whole minute, the value is displayed in seconds, and the suffix SECONDS is displayed.

Response 5 (PARTITION):

If QUERY MONITOR PARTITION is specified, the response is:

```
MONITOR PARTITION n
```

**PARTITION *n***

indicates the number of 4 KB pages allocated to hold for event configuration and data records.

Response 6 (RATE):

If QUERY MONITOR RATE is specified, the response is:

```
MONITOR RATE {m SECONDS} [PENDING RATE {n SECONDS}]
             {STOP      }             { STOP  }]
```

**RATE *n* SECONDS**

**RATE STOP**

describes the current monitor high-frequency sampling rate, where *n* is a value between 0.01 and 30 seconds. STOP indicates that high-frequency sampling is inactive.

**PENDING RATE *n* SECONDS**

**PENDING RATE STOP**

indicates that a new high-frequency sampling rate is to take effect after the current high-frequency sampling rate has occurred, where *n* is a value between 0.01 and 30 seconds. STOP indicates that high-frequency sampling becomes inactive when the current RATE expires.

Response 7 (SAMPLE):

If QUERY MONITOR SAMPLE is specified, the response is:

```

MONITOR SAMPLE {ACTIVE      } {NEXT date time}
                {INACTIVE   }
                {SUSPENDED  }
                {PENDING    }
                INTERVAL {n MINUTES} [PENDING INTERVAL {n MINUTES}]
                {n SECONDS} [ {n SECONDS}]
                RATE     {n SECONDS} [PENDING RATE     {n SECONDS}]
                {STOP     } [ {STOP     } ]

MONITOR DCSS NAME - {dcssname      }
                   {NO DCSS NAME DEFINED}

CONFIGURATION  SIZE n          LIMIT {n MINUTES}
                                           {n SECONDS}

CONFIGURATION AREA {WILL BE FREED WITHIN - {n MINUTES n SECONDS}}
                  {IS FREE}

USERS CONNECTED TO *MONITOR {NO USERS CONNECTED
                             {userid[QUIESCED][PROCESSING-CONFIG][EXCLUSIVE]}
                             [PENDING-CONFIG ]

```

**SAMPLE**

describes the status of sample monitoring.

**ACTIVE**

indicates that SAMPLE monitoring is being performed because a MONITOR START or MONITOR SAMPLE START command was issued and there are eligible monitor users to receive sample data notification messages.

**INACTIVE**

indicates that SAMPLE monitoring is not being performed because no monitor start command for a sample has been issued.

**SUSPENDED**

indicates that SAMPLE monitoring is suspended because monitor is running in exclusive mode and the user connected is quiesced. (If the user is quiesced before the start command is issued, the monitor is in a PENDING state.)

**PENDING**

indicates that SAMPLE monitoring is not being performed because there is no eligible monitor user to receive data notification for sample data.

**NEXT *date time***

for ACTIVE, SUSPENDED, or PENDING, reports the date and time that the next sample collection will begin, if SAMPLE is ACTIVE at that time.

**INTERVAL *n* MINUTES**

**INTERVAL *n* SECONDS**

describes the current monitor sampling interval, where *n* is the current interval value (1 through 60 for minutes, 6 through 3600 for seconds). If an INTERVAL value does not fall on a whole minute, the value is displayed in seconds, and the suffix SECONDS is displayed.

### **PENDING INTERVAL *n* MINUTES**

### **PENDING INTERVAL *n* SECONDS**

indicates that a new sampling interval is to take effect after the current sampling interval has occurred. If the new INTERVAL value does not fall on a whole minute, the value is displayed in seconds, and the suffix SECONDS is displayed.

### **RATE *n* SECONDS**

### **RATE STOP**

describes the current monitor high-frequency sampling rate, where *n* is a value between 0.01 and 30 seconds. STOP indicates that high-frequency sampling is inactive.

### **PENDING RATE *n* SECONDS**

### **PENDING RATE STOP**

indicates that a new high-frequency sampling rate is to take effect after the current high-frequency sampling rate has occurred; *n* is a value between 0.01 and 30 seconds. STOP indicates that high-frequency sampling is inactive when the current RATE expires.

### **MONITOR DCSS NAME**

provides the name of the DCSS that is to collect monitor data.

#### **dcssname**

is the name of the DCSS being used.

### **NO DCSS NAME DEFINED**

indicates that no DCSS name is defined.

### **CONFIGURATION SIZE *n***

indicates the number of 4 KB pages (within the sample portion of the DCSS) allocated for sample configuration records. Leading zeros, if any, are removed.

### **LIMIT *n* MINUTES**

### **LIMIT *n* SECONDS**

displays the time limit in which a reply is required to the configuration IUCV message. If the value does not fall on a whole minute, the value is displayed in seconds, and the suffix SECONDS is displayed. Leading zeros, if any, are removed.

### **CONFIGURATION AREA WILL BE FREED WITHIN *-n* MINUTES *n* SECONDS**

displays the amount of time that the users holding the configuration area have left to reply to the configuration IUCV message.

### **CONFIGURATION AREA IS FREE**

is displayed if no users are holding the configuration area.

### **USERS CONNECTED TO \*MONITOR**

lists, in alphabetic order, the user IDs of the users connected to \*MONITOR to collect SAMPLE data.

### **NO USERS CONNECTED**

indicates that no user is connected to \*MONITOR to collect sample data.

#### **userid**

identifies the users connected to \*MONITOR for SAMPLE data.

### **QUIESCED**

indicates that the user is not allowing \*MONITOR to send IUCV messages over this path to the user's virtual machine.

### **PROCESSING-CONFIG**

indicates that the user received the notification for the configuration send, but the user has not yet replied to the send.

### **PENDING-CONFIG**

indicates that the user is waiting for a configuration notification message. The user is not eligible for data notification until after a configuration message has been successfully sent.

**Note:** If a user's path is at the IUCV message limit, the user remains quiesced) until the user responds to previous messages from \*MONITOR.

**EXCLUSIVE**

indicates that monitor is running in exclusive mode, not shared mode. The mode is specified by the format of the connect interface used by the first user that connects to \*MONITOR.

The following additional response lines are given based on the specifications in the sample profile:

```
MONITOR  DOMAIN  ENABLED
SYSTEM   DOMAIN  ENABLED

PROCESSOR DOMAIN  {DISABLED      }
          {ENABLED CPUMFC }
          {ENABLED NOCPUMFC}

STORAGE  DOMAIN  {DISABLED}
          {ENABLED }

USER     DOMAIN  {DISABLED}
          {ENABLED }
```

**Note:** The following additional responses are issued only if the USER domain is enabled, based on the specifications for the USER domain in the sample profile:

```
ALL USERS ENABLED
```

- or -

```
ALL USERS EXCEPT:
  userid [userid...]
```

- or -

```
THE FOLLOWING USERIDS ARE ENABLED:
  userid [userid...]
```

```
I/O      DOMAIN  {DISABLED}
          {ENABLED }
```

**Note:** The following additional responses are issued only if the I/O domain is enabled, based on the specifications for the I/O domain in the sample profile:

```
PCIF CLASS ENABLED
```

- or -

```
PCIF CLASS DISABLED
```

- and -

```
ALL DEVICES ENABLED
```

- or -

```
ALL DEVICES EXCEPT:
  rdev[-] or rdev
```

- or -

```
THE FOLLOWING DEVICES ARE ENABLED:
  rdev[-rdev] or [rdev... rdev]
```

```
APPLDATA DOMAIN  {DISABLED}
          {ENABLED }
```

**Note:**

1. For the APPLDATA domain, a user is enabled only if the OPTION APPLMON statement is specified in its directory entry.

2. The following additional responses are issued only if the APPLDATA domain is enabled, based on the specifications for the APPLDATA domain in the sample profile:

```
ALL USERS ENABLED
```

- OR -

```
ALL USERS EXCEPT:  
  userid [userid...]
```

- OR -

```
THE FOLLOWING USERIDS ARE ENABLED:  
  userid [userid...]
```

```
NETWORK    DOMAIN {DISABLED}  
            {ENABLED }
```

```
ISFC       DOMAIN {DISABLES}  
            {ENABLED }
```

```
SSI        DOMAIN {DISABLED}  
            {ENABLED }
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}



## QUERY MSS

► Query — MSS ◄

### Authorization

Privilege Class: B

### Purpose

Use QUERY MSS to display the status of multiple subchannel set support and its configuration.

### Usage Notes

1. The *active* configuration is the set of accessible devices currently being used by or available for use by the z/VM system and virtual machines. The *standby* configuration is the set of accessible devices not currently available for use. For example, if the system was IPLed from a device in subchannel set 0, all devices in that subchannel set would be part of the active configuration while devices in other configured subchannel sets would be part of the standby configuration. If a HYPERSWAP command was used to swap some devices in subchannel set 0 with counterparts in another subchannel set, the swapped subchannel set 0 devices would become part of the standby configuration and the corresponding devices in the swapped subchannel set would become part of the active configuration.
2. Last HYPERSWAP contains the date and time of the last successful HYPERSWAP, the userid which issued the HYPERSWAP command and the source, and target subchannel set of the HYPERSWAP.

### Responses

Response 1:

If multiple subchannel set support is not enabled, you will receive the following response:

```
MSS Disabled
```

Response 2:

If multiple subchannel set support is enabled with subchannel sets 0-3 configured, the system was IPL'd from a device in SS0 and a HYPERSWAP has occurred from subchannel set 0 to subchannel set 1, you will receive a response that looks like this:

```
MSS Enabled
Subchannel Sets Configured: 0, 1, 2, 3
Subchannel Set of IPL Device: 0
Last HYPERSWAP: 2017-08-23 13:19:04, Userid = USERG, Source = SS0, Target = SSI
```

Response 3:

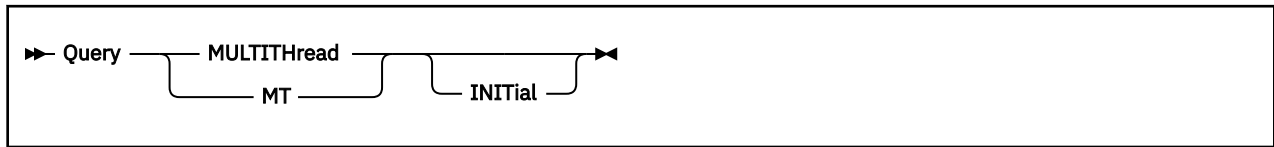
If multiple subchannel set support is enabled with subchannel sets 0-1 configured, the system was IPL'd from a device in SS0 and a HYPERSWAP has not occurred, you will receive a response that looks like this:

```
MSS Enabled
Subchannel Sets Configured: 0, 1
Subchannel Set of IPL Device: 0
Last HYPERSWAP: None
```

### Messages

- HCP003E Invalid option - *option*

## QUERY MULTITHREAD



### Authorization

Privilege Class: A, B, C, E, G

### Purpose

Use QUERY MULTITHREAD to display the system's multithreading (MT) status and thread information.

### Operands

#### INITIAL

This optional operand requests that the initial configuration values specified on the MULTITHREADING statement in the system configuration file be displayed with the resulting effective configuration values.

If INITIAL is omitted the current configuration settings are displayed.

### Usage Notes

1. For information about configuring multithreading, see the MULTITHREADING system configuration statement in *z/VM: CP Planning and Administration*. To display processor core utilization information, use the INDICATE MULTITHREAD command.  
Set the “[SET MULTITHREAD](#)” on page 1578 command for information about how to modify the multithreading configuration.
2. The multithreading configuration information displayed by the QUERY MULTITHREAD command is the current configuration resulting from the MULTITHREADING statement, or as modified by a successful SET MULTITHREAD command.
3. If Q MULTITHREAD is issued while SET MULTITHREAD processing is active the command will respond that the multithreading configuration is in transition. This will be indicated by message HCP6513I and corresponding return code.
4. In order to delay processing in an automation script until the asynchronous processing for a previously issued SET MULTITHREAD command is complete, the script can loop (with a delay) on QUERY MULTITHREAD until the return code no longer indicates SET MULTITHREAD is in progress. The INITIAL operand should not be specified for this use because it completes with a zero return code even if a configuration change is in progress.
5. The system can be restored to the initial values by use of the SET MULTITHREAD INITIAL command.

### Responses

Response 1:

```
The Multithreading Facility is not installed.
```

is displayed if the multithreading facility is not installed on the hardware or the logical partition, or if the z/VM system is running as a guest.

Response 2:

```
Multithreading is disabled.
```

is displayed if the processor or configuration supports multithreading but multithreading is disabled. There are a number of reasons why multithreading might be disabled. For example, a MULTITHREADING statement might not be included in the system configuration file. Review the system initialization messages in the operator's console log.

Response 3:

```
Multithreading is enabled.
      Requested   Activated
      Threads     Threads
MAX_THREADS      nn         nn
CP core          nn         nn
IFL core         nn         nn
ICF core         nn         nn
zIIP core        nn         nn
```

This response is displayed if the processor or configuration supports multithreading and multithreading is enabled in the system configuration file.

**Requested Threads**

This column indicates the specified or default value from the associated operand on the MULTITHREADING configuration statement, or as modified by a successful SET MULTITHREAD command. If the operand was not specified or the MAX keyword was specified, this column will show the string "MAX".

**Activated Threads**

For MAX\_THREADS, the activated threads number is the least of the following:

- The value specified for the MAX\_THREADS operand on the MULTITHREADING statement
- The highest number of threads per core supported for any CPU type that is available on the hardware or the configuration where z/VM is running
- The highest number of threads per core supported by z/VM for any CPU type

The activated threads value for MAX\_THREADS, rounded up to a power of 2, is the number of CPU addresses assigned to each core.

For each CPU type, the activated threads number is the least of the following:

- The activated threads value for MAX\_THREADS
- The requested threads value for that CPU type on the MULTITHREADING statement, or as modified by a successful SET MULTITHREAD command.
- The number of threads per core supported for that CPU type on the hardware or the configuration where z/VM is running
- The number of threads for that CPU type supported by z/VM

The activated threads value for each CPU type is the number of threads available for use by CP in an online core of the specified CPU type. Activated threads appear as processors to z/VM. If this number is less than the activated MAX\_THREADS value, or if the activated MAX\_THREADS value is not a power of 2, then there will be gaps in the range of online CPU addresses.

Response 4:

```
Multithreading is enabled and these are the initial values
      Requested   Activated
      Threads     Threads
MAX_THREADS      nn         nn
CP core          nn         nn
IFL core         nn         nn
ICF core         nn         nn
zIIP core        nn         nn
```

This response is displayed when the INITIAL operand is specified, if the processor or configuration supports multithreading and multithreading is enabled in the system configuration file. These are the initial values which were in effect at system initialization which may have since been modified by a SET MULTITHREAD command.

## QUERY MULTITHREAD

See response 3 for an explanation of the output with the following differences in mind. This response describes the multithreading configuration resulting from the requested configuration on the MULTITHREADING configuration statement. The current multithreading configuration may have been changed by a SET MULTITHREAD command.

### Examples

If the ENABLE operand was specified on the MULTITHREADING statement in the system configuration file, and you issue the QUERY MULTITHREAD and INDICATE MULTITHREAD commands on a system that is capable of multithreading, you might see responses similar to the following example. Note that gaps might occur in the sequence of processor addresses for cores that have less than the maximum number of activated threads.

**Note:** This example is for z/VM 6.3, which supports up to 2 threads on an IFL core and 1 thread on a core of any other type.

```
query mt
Multithreading is enabled.
           Requested  Activated
           Threads   Threads
MAX_THREADS  MAX      2
CP core      MAX      1
IFL core     MAX      2
ICF core     MAX      1
zIIP core    MAX      1
```

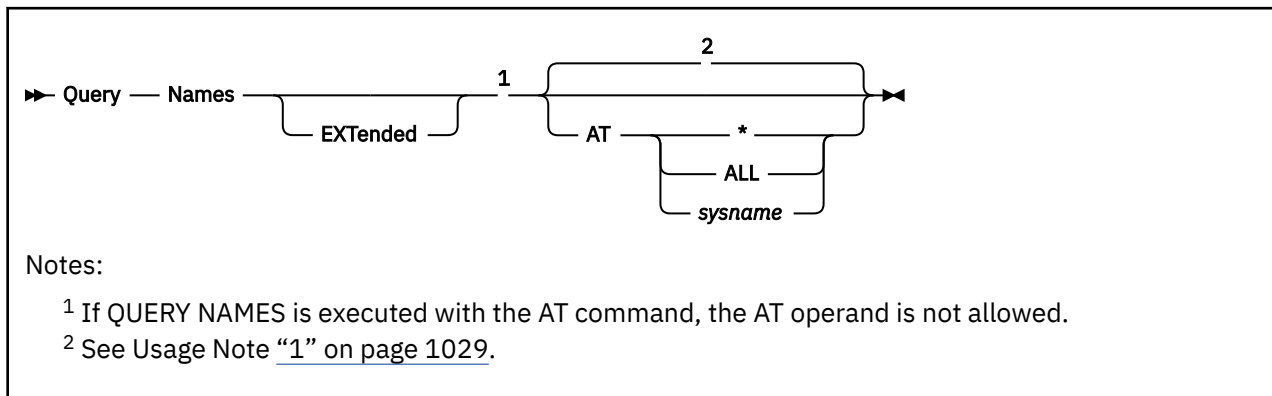
```
query proc
PROCESSOR 00 MASTER CP CORE 0000
PROCESSOR 02 ALTERNATE CP CORE 0001
PROCESSOR 04 ALTERNATE CP CORE 0002
PROCESSOR 06 ALTERNATE IFL CORE 0003
PROCESSOR 07 ALTERNATE IFL CORE 0003
PROCESSOR 08 ALTERNATE IFL CORE 0004
PROCESSOR 09 ALTERNATE IFL CORE 0004
PROCESSOR 0A ALTERNATE IFL CORE 0005
PROCESSOR 0B ALTERNATE IFL CORE 0005
PROCESSOR 0C ALTERNATE ZIIP CORE 0006
PROCESSOR 0E ALTERNATE ZIIP CORE 0007
PROCESSOR 10 ALTERNATE CP CORE 0008
```

```
indicate mt
Multithreading is enabled.
Statistics from the interval 13:00:30 - 13:01:00
Core Type CP   Busy 8%   TD 1.00 of 1   Prod 100%   Util 8%
CF 100%   MaxCF 100%
Core Type IFL  Busy 67%   TD 1.24 of 2   Prod 82%   Util 55%
CF 107%   MaxCF 131%
Core Type ZIIP Busy 5%   TD 1.00 of 1   Prod 100%   Util 5%
CF 100%   MaxCF 100%
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP6513I A multithreading configuration change is in progress.

## QUERY NAMES



### Authorization

Privilege Class: A, B, C, D, E, F, G

### Purpose

Use QUERY NAMES to display:

- A list of all logged-on users
- The real or logical device number of the display to which each user is connected

### Operands

#### EXTENDED

specifies that logical unit network qualifiers or logical device host names and IP addresses are displayed if they exist. See [Response 2](#) for details on how the EXTENDED operand changes the responses.

#### AT \*

displays logged-on users for the system you are logged on to and the real or logical device number of the displays where those users are connected. This is the default if the system is not a member of an SSI cluster.

#### AT ALL

displays logged-on users for all systems in the SSI cluster and the real or logical device number of the displays where those users are connected. A response is issued for each active member of the cluster. Responses are displayed in the order they are received, and are prefixed with the source member name.

#### AT sysname

displays logged-on users for a specific system in the SSI cluster and the real or logical device number of the displays where those users are connected.

### Usage Notes

1. If the AT operand is not specified:

- If QUERY NAMES is not executed with the AT command:
  - If the issuer's system is not a member of an SSI cluster, the default is AT \*.
  - If the issuer's system is a member of an SSI cluster, the command displays all users defined by single-configuration virtual machine definitions logged on to any member of the cluster, and local logon instances of users defined by multiconfiguration virtual machine definitions.

## QUERY NAMES

If you want to display all logon instances of users defined by multiconfiguration virtual machine definitions, issue QUERY NAMES AT ALL. If you want to display only the users on the local system, issue QUERY NAMES AT \*.

- If QUERY NAMES is executed with the AT command, the AT operand function is bypassed (there is no default), and information is displayed for the system specified on the AT command.
2. Multiple, concurrent VTAM service machines are supported; therefore, multiple listings of logical unit names (*lunames*) owned by VTAM service machines might be displayed in the response. This portion of the QUERY NAMES response follows the *rdev/ldev* listings of users connected to the system; it is displayed only if there is an active VTAM service machine connected to the system.
3. Within a single VSM that supports network qualified names, multiple listings of logical unit names (*lunames*) owned by VTAM service machines might be displayed in the response, if EXTENDED is not specified.
4. If a large virtual machine is logged off multiple times and two or more logged-off instances are still going through memory reset, the user ID (transformed from uppercase to lowercase) will appear multiple times in the responses to this command. Eventually, each instance will finish memory reset and will disappear from the responses.

The following sequence of events is an example of this type of situation:

- a. A user logs on virtual machine "LARGEONE" and runs a workload that instantiates a large amount of memory.
- b. The user logs off the virtual machine. This instance of virtual machine "LARGEONE" becomes known as instance "largeone."
- c. Responses to QUERY NAMES include "largeone - LOF."
- d. The user logs back on to virtual machine "LARGEONE" and runs a workload that instantiates a large amount of memory.
- e. The user logs off. This instance of virtual machine "LARGEONE" also becomes known as instance "largeone."
- f. Responses to QUERY NAMES now include two occurrences of "largeone - LOF."
- g. Eventually instance "largeone" from Step "4.b" on page 1030 completes reset and disappears.
- h. Responses to QUERY NAMES now include only one occurrence of "largeone - LOF."
- i. Eventually instance "largeone" from Step "4.e" on page 1030 completes reset and disappears.

### Examples

Example 1:

When you issue QUERY NAMES to display logged-on users to the SSI cluster, you might get a response similar to this:

```
0001      - SSI , 0002      - SSI
OPERATOR  - 2000
```

Example 2:

The source member name is prefixed in an SSI cluster when you specify Q N AT ALL.

```
PERRY2    : 0002      - DSC ,0001      - DSC , OPERATOR - 2000
PERRY3    : OPERATOR - 2000
```

### Responses

Response 1:

```
{userid } - {SSI }
{userid } - {rdev}
{userid } - {LOF }
```

... up to four responses per line

```
{LOGLxxxx}   {ldev}
{LOGNxxxx}   {DSC }
{LOGVxxxx} - luname}
{LOGNMxxx} - DSC}
{LOGNSYSC}   {SYSC}
{LOGNSYSG}   {SYSG}
{lognauto}   {DSC }
[VSM - userid
 [userid - luname, userid -luname,... ]
 [userid - luname, userid -luname,... ]
```

**userid**

**LOGNxxxx**

**LOGLxxxx**

**LOGVxxxx**

**LOGNMxxx**

**LOGNSYSC**

**LOGNSYSG**

**lognauto**

identifies the virtual machine. If a user has not completed logon processing to a real device, *userid* is displayed in the form LOGNxxxx, where xxxx is the real device number of the display being used. If a user has not completed logon processing to a logical device, *userid* is displayed in the form LOGLxxxx, where xxxx is the logical device number of the display being used. If a user has not completed logon processing to an SNA/CCS terminal, *userid* is displayed in the form LOGVxxxx, where xxxx is a unique identifier. If a user is in the process of being relocated on to this system from another member of the SSI cluster, LOGNMxxx is displayed, where xxx is a unique value. If a user has not completed logon processing to the system console, user ID LOGNSYSC is displayed. If a user has not completed logon processing to the integrated 3270 console, user ID LOGNSYSG is displayed. If a user has not completed AUTOLOG or XAUTOLOG processing, the *userid* is displayed as lognauto. There might be multiple instances of lognauto for different users.

**xxx and xxxx**

can be any character, number, or symbol. Userids starting with LOGN, LOGL, and LOGV are restricted.

**rdev**

**ldev**

**DSC**

**SYSC**

**SYSG**

**SSI**

identifies the device number where the user is logged on. The device may be real (*rdev*) or logical (*ldev*). DSC is displayed if the user is disconnected and does not have a real display. SYSC is displayed if the user is logged on to the system console. SYSG is displayed if the user is logged on to the integrated 3270 console. SSI is displayed if the user is logged on to a remote member of the SSI cluster.

**LOF**

indicates that the user has logged off but its memory is still in the process of being released.

**VSM - userid**

identifies the user ID of the VTAM service machine managing the following users.

**userid - luname**

identifies the user IDs and the SNA logical unit names (*lunames*) of their terminals for all users accessing the system from an SNA/CCS terminal managed by that VTAM service machine.

**Response 2:**

```
userid - ldev HOST hostname [FROM IP-address]
userid - netname.luname
```

## QUERY NAMES

### **HOST *hostname***

identifies the virtual machine that created and manages the logical device where the user is logged in.

### **FROM IP-*address***

identifies the IP address associated with the logical device.

An IPv4 address is displayed in dotted-decimal format.

**Example:** 9 . 130 . 44 . 144

An IPv6 address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros).

**Example:** The following is an IPv6 address:

```
F802:0000:0000:0000:0000:0A00:0001
```

This address is displayed in compressed format as:

```
F802::A00:1
```

### ***userid - netname.luname***

identifies the user IDs and the fully qualified SNA logical unit names (*netname.luname*) of their terminals for all users accessing the system from an SNA/CCS terminal managed by that VTAM service machine.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP1142E The AT option is not allowed when this command is executed remotely with the AT command.
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.



## QUERY NEW\_DEVICES

► Query — NEW\_DEVICES ◄

### Authorization

Privilege Class: C

### Purpose

Use QUERY NEW\_DEVICES to display whether CP initializes new real devices as they are added to a running system.

### Usage Notes

1. To add a new device to a running system, use the DEFINE DEVICE command (see [“DEFINE DEVICE / IODEVICE”](#) on page 167).
2. If CP does not initialize a new device when you add it to the system, you must vary the device online using the VARY (real device) command (see [“VARY \(Real Device\)”](#) on page 2063) before anyone can use that device.
3. To change the way CP initializes new real devices as they are added to a running system:
  - During initialization, use the ENABLE or DISABLE NEW\_DEVICES\_INITIALIZED\_WHEN\_ADDED operands of the FEATURES Statement in your system configuration file. For more information, see *z/VM: CP Planning and Administration*.
  - After initialization, use the SET NEW\_DEVICES command (see [“SET NEW\\_DEVICES”](#) on page 1581).
4. For more information about adding new devices to a running system, see the [FEATURES Statement](#) in *z/VM: CP Planning and Administration*.

### Responses

Response 1 (Devices Are Initialized):

To display how CP handles new real devices as they are added to your running system, enter the following:

```
query new_devices
New devices are initialized when added.
Ready;
```

This response tells you that any new device you add will be immediately available for use.

Response 2 (Devices Are Not Initialized):

If CP is not initializing new devices as they are added, you would see:

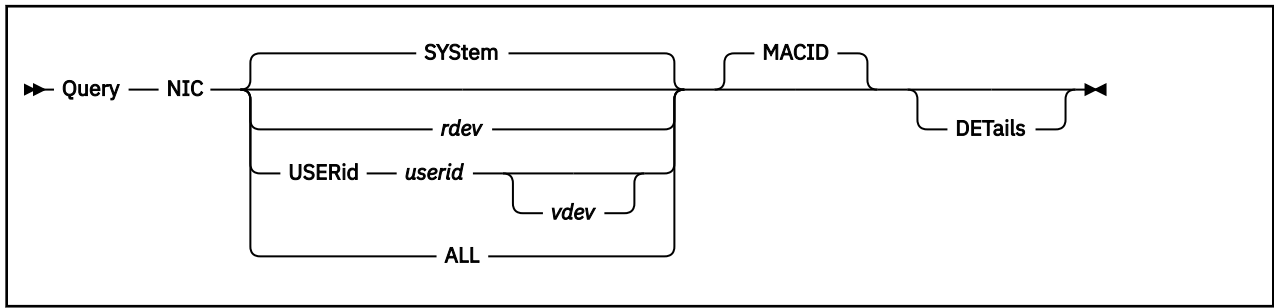
```
query new_devices
New devices are not initialized when added.
Ready;
```

This response tells you that new devices must be varied online before they become available for use.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*

## QUERY NIC



### Authorization

Privilege Class: B

### Purpose

Use QUERY NIC to display the system-level information and the MAC addresses assigned or being used by the selected network devices.

### Operands

#### SYSTEM

display the current system level information.

#### rdev

is the real network device number on this system for which information will be returned. RDEV information is only displayed when a MACID has been set.

#### USERid \*|USERid userid vdev

is the user ID of the virtual machine and virtual network device number for which you want to query information. If *userid* is specified as an asterisk (\*), CP uses the user ID of the virtual machine issuing the command.

#### vdev

is the virtual network device number for which information will be returned. If a *vdev* is not specified, all virtual network devices for the specified user ID will be returned.

#### ALL

will display information for the system and all network devices.

#### MACID

to display MAC addresses assigned or being used by the selected network devices.

When running z/VM as a single system image cluster, the command will also display MAC addresses originally assigned by this system that were moved by Live Guest Relocation to another member of the single system image cluster. The query will not display all the MAC addresses assigned within a single system image cluster. It will only display MAC addresses originally assigned by this system.

#### DETAils

to display detailed information about the selected network devices.

### Usage Notes

The SET NIC command is used to modify NIC settings.

### Examples

1. QUERY NIC:

Display the system level information:

```
q nic
Default System MAC Protection: OFF
Network Device Allocation: Permitted
```

2. QUERY NIC ALL MACID DETAILS:

Display the system level information and all MAC addresses being used by the network devices:

```
q nic all macid details
Default System MAC Protection: ON
Network Device Allocation: Permitted
  MAC: 02-22-90-00-00-00 SYSTEM 0000
  MAC: 02-22-90-00-00-01 SYSTEM ETH1
A MAC: 02-22-90-00-00-02 TCPIP 0600
  Protection: Unspecified
A MAC: 02-22-90-00-00-03 TCPIP 4100
  Protection: OFF
  MAC: 02-22-90-00-00-04 SYSTEM ETH2
  Protection: Unspecified
R MAC: 02-22-FF-00-03-21 TCPIP 0600 ON 2NDLVLF
R MAC: 02-22-FF-00-9C-08 OPERATOR 9C08 ON 2NDLVLF
R MAC: 02-22-FF-01-42-00 LINUX3 7000 ON 2NDLVLF
W MAC: 02-55-90-00-03-21 TCPIP 0600
  Protection: Unspecified
  MAC: 02-55-90-00-9C-08 OPERATOR 9C08 ON 9C08
PA MAC: 02-55-90-01-42-00 LINUX3 7000
  Protection: ON
```

3. A VIRTUAL version of this command, QUERY VIRTUAL NIC, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL NIC” on page 1310](#).

**Responses**

Response 1:

The QUERY NIC response includes optional lines. The general format is:

```
Default System MAC Protection: level
Network Device Allocation: state
ps MAC: mm-mm-mm-mm-mm userid vdev/switchname ON rdev/system
  Protection: setting
```

**level**

Default system MAC address protection

**ON**

specifies the MAC address is protected and cannot be overridden by the guest when establishing a network connection.

**OFF**

specifies the MAC address is not protected. A guest can override an assigned MAC address when establishing a network connection.

The system MAC address protection setting is used when MAC address protection is set to UNSPECified for a specific device, a coupled virtual switch or guest LAN

**state**

Network Device Allocation: *state*

**Permitted**

The SSI cluster is in a STABLE mode. While the SSI cluster is STABLE, HiperSocket or OSA type network devices may either be created or attached to a guest's virtual I/O configuration.

**Prohibited**

The SSI cluster is currently NOT in a STABLE mode. Creating or attaching a network device to a guest's virtual I/O configuration will not be allowed in order to maintain network integrity. The CP DEFINE NIC and ATTACH commands issued for a HiperSocket or OSA device will fail until the SSI cluster returns to a STABLE mode. The CP QUERY SSI command may be used to determine the reason the SSI cluster is not in a STABLE mode.

**ps**

MAC Address Attributes

**P**

In the *p* column indicates the MAC address is protected and cannot be overridden by the guest when establishing a network connection. If *p* is not indicated, then the MAC address is not protected.

**R**

in the *p* column indicates the MAC address is remote and specifies that the MAC address is currently assigned to a network device, adapter or virtual switch in another member of an SSI cluster.

**Note:** When displaying a remote MAC address, no other MAC address attributes will be displayed.

**W**

in the *s* column indicates the MAC address is waiting to be assigned the next time a network connection is reestablished.

**A**

in the *s* column indicates the MAC address is assigned to a virtual NIC adapter. This is the default MAC address that will be used when establishing a network connection on a device that doesn't have a MAC address assigned by a SET NIC MACID CP command.

**MAC: mm-mm-mm-mm-mm-mm**

MAC address assigned by this system.

**userid**

is the user ID that owns the MAC address.

**vdev/switchname**

is either the four digit virtual device number or the eight character virtual switch name assigned to the MAC address. When an entry is displayed for a MAC address assigned to a virtual switch that is being used by another system in the SSI, *vdev* will be a pseudo device number. In this case, the device number can be ignored.

**ON rdev/system**

is either the four digit real device number for a MAC address assigned by the local system or the eight character system name of another system in the single system image cluster that is using a MAC address assigned by this system. This information will only be displayed if there is a real device associated with the network device or the MAC address assigned by this system is being used by another system in the single system image cluster.

A system name of "Unknown" could indicate a problem with the single system image cluster.

**setting**

MAC address protection

**ON**

specifies the MAC address is protected and can't be overridden by the guest when establishing a network connection.

**OFF**

specifies the MAC address is not protected. A guest can override an assigned MAC address when establishing a network connection.

**Unspecified**

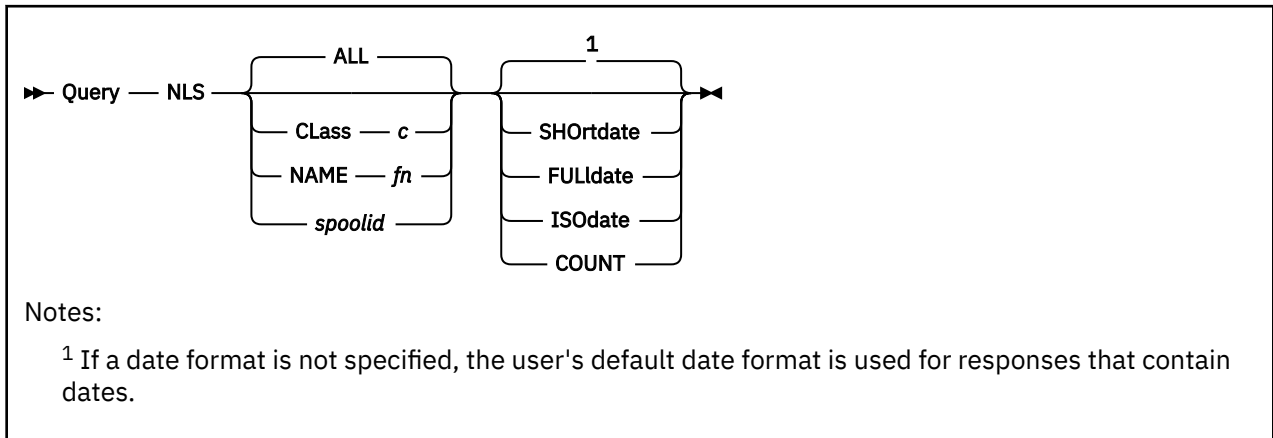
specifies the MAC address is not protected. A guest can override an assigned MAC address when establishing a network connection.

This is the device MAC protection setting as specified with the SET NIC command.

**Messages**

- HCP003E Invalid option - command contains extra option(s) starting with xxx
- HCP2768E Missing user ID
- HCP2836E [RDEV | VDEV | USERID] was not found.
- HCP6706E Invalid user ID - xxxxxxxx

## QUERY NLS



### Authorization

Privilege Class: E

### Purpose

Use QUERY NLS to display:

- Message repository file information
- The count of message repository files

**Note:** Message repository files are created by the system programmer.

### Operands

#### ALL

queries all message repository files. This is the default.

#### Class *c*

specifies the class of files to be queried. The operand *c* can be either A or P, where A indicates that the repository file is in the available state and P indicates that the file is in the pending purge state.

#### NAME *fn*

specifies the file name of the file or files to be queried. This is the file name specified when the file was created. More than one file can exist with the same file name, although only one may be class A.

#### *spoolid*

specifies the system-assigned number of the file to be queried.

#### SHOrtdate

specifies that dates in the response to this command will be displayed in *mm/dd* format, where *mm* is the month and *dd* is the day of the month.

#### FULldate

specifies that dates in the response to this command will be displayed in *mm/dd/yyyy* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

#### ISOdate

specifies that dates in the response to this command will be displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

#### COUNT

displays only the number of message repository files that match the criteria specified on the command.

## Usage Notes

1. The default date format for certain CP and CMS commands can be set on a system-wide basis and also for the individual user. The system-wide default date format is set with the SYSTEM\_DATEFORMAT system configuration statement. The user's default date format is set with the DATEFORMAT directory statement. The system-wide default and the user's default can also be set with the CP SET DATEFORMAT command. The user's default date format defaults to the system-wide default. The system-wide and user settings can be queried with the CP QUERY DATEFORMAT command. The hierarchy of possible date format settings for the QUERY NLS command, from highest priority to lowest, is:
  - QUERY NLS command operand
  - User default
  - System-wide default

## Responses

Response 1:

The following response is issued when you specify QUERY NLS SHORTDATE:

OWNERID	FILE	TYPE	CL	RECS	DATE	TIME	FILENAME	FILETYPE	ORIGINID
*NLS	nnnn	type	c	nnnn	mm/dd	hh:mm:ss	fn	ft	originid

Response 2:

The following response is issued when you specify QUERY NLS FULLDATE:

OWNERID	FILE	TYPE	CL	RECS	DATE	TIME	FILENAME	FILETYPE	ORIGINID
*NLS	nnnn	type	c	nnnn	mm/dd/yyyy	hh:mm:ss	fn	ft	originid

Response 3:

The following response is issued when you specify QUERY NLS ISODATE:

OWNERID	FILE	TYPE	CL	RECS	DATE	TIME	FILENAME	FILETYPE	ORIGINID
*NLS	nnnn	type	c	nnnn	yyyy-mm-dd	hh:mm:ss	fn	ft	originid

### OWNERID

indicates that the file is an NLS file owned by the system. The designation \*NLS is not a user ID. It indicates the queue on which the file resides.

### FILE

identifies the spool file.

### TYPE

identifies the type of system data file. In this case, it is NLS and indicates a message repository file.

### CL

identifies the file class, either A or P. Class A indicates that the file is available. Class P indicates that the file is in the pending purge state; it is purged during the next system IPL or RESTART.

### RECS

identifies the number of 4 KB pages in the file.

### DATE

identifies the date the file was closed. The format of the date is determined by the SHORTDATE, FULLDATE, or ISODATE operand. If no date format operand is specified, the user's default date format setting is used.

### TIME

identifies the time the file was closed in hours, minutes, and seconds.

### FILENAME

identifies the file name associated with the file.

## QUERY NLS

### FILETYPE

identifies the file type associated with the file.

### ORIGINID

identifies the originator of the file.

Response 4:

```
FILES:  NO NLS
```

appears if none of the specified files exist.

Response 5:

```
FILES:  {nnnn} NLS  
        { NO }
```

appears if the COUNT option is specified on the QUERY NLS command. Only the number of repository files that match the specified criteria is displayed.

### nnnn

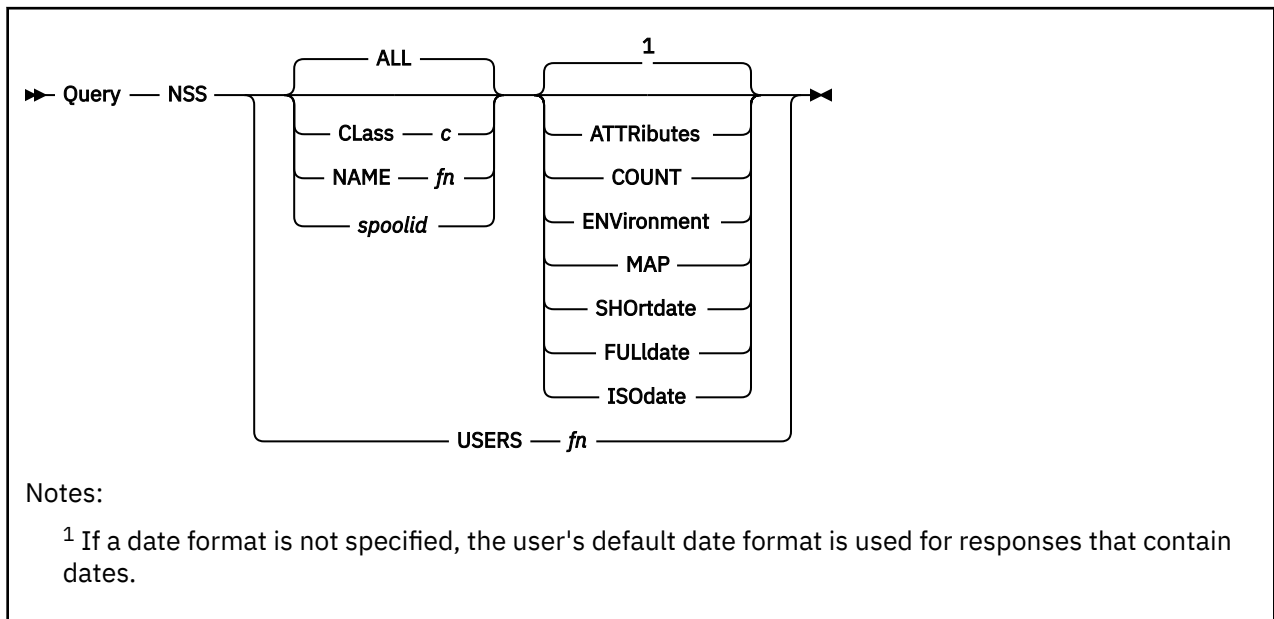
identifies the number of repository files. If no files are found that match the selection criteria, NO replaces nnnn.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP849E A user ID must be specified with the spoolid
- HCP1558E Command processing cannot complete.



## QUERY NSS



### Authorization

Privilege Class: E

### Purpose

Use QUERY NSS to display information about system data files that contain named saved systems (NSSs) and saved segments.

### Operands

#### ALL

requests information on all NSS and saved segment files. This is the default.

#### Class *c*

specifies the class of files to be queried. The operand *c* can be one of the following:

#### A

Available nonrestricted.

#### P

Pending purge.

#### R

Available restricted.

#### S

Skeleton.

For further definitions of these classes, see "Response 7 (MAP)" under ["Responses" on page 1042](#).

#### NAME *fn*

specifies the file name of the file for which information is requested.

#### *spoolid*

specifies the spool file number of the file to be queried.

#### ATTRIBUTES

displays additional information specific to NSS and saved segment files.

**COUNT**

displays only the number of files that match the criteria specified on the command. No other information is provided.

**ENVIRONMENT**

for NSS files only, displays the information about the IPL environment, such as the minimum virtual machine size and the allowed virtual machine modes.

**MAP**

indicates that information specific to NSS and saved segment files should be displayed, such as beginning and ending page numbers of the saved NSS and saved segments.

**SHORTDATE**

specifies that dates in the response to this command will be displayed in *mm/dd* format, where *mm* is the month and *dd* is the day of the month.

**FULLDATE**

specifies that dates in the response to this command will be displayed in *mm/dd/yyyy* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

**ISODATE**

specifies that dates in the response to this command will be displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

**USERS *fn***

requests a list of the user IDs that are actively using the NSS or saved segment.

**Usage Notes**

1. QUERY NSS is equivalent to the QUERY RDR/PRT/PUN command for spool files.
2. The QUERY NSS ALL MAP command does not reflect all associations of a member that is within multiple segment spaces. QUERY NSS NAME *spacename* MAP should be used to determine the members of a particular segment space.
3. For security purposes, if the issuing user is not authorized to use a system data file, then the MAP, ENV and ATTR options of the QUERY NSS command will not provide information on those system data files.
4. The default date format for certain CP and CMS commands can be set on a system-wide basis and also for the individual user. The system-wide default date format is set with the SYSTEM\_DATEFORMAT system configuration statement. The user's default date format is set with the DATEFORMAT directory statement. The system-wide default and the user's default can also be set with the CP SET DATEFORMAT command. The user's default date format defaults to the system-wide default. The system-wide and user settings can be queried with the CP QUERY DATEFORMAT command. The hierarchy of possible date format settings for the QUERY NSS command, from highest priority to lowest, is:
  - QUERY NSS command operand
  - User default
  - System-wide default
5. Processing of a QUERY NSS command might be delayed if DIAGNOSE code X'64' is concurrently performing the initial load of a large DCSS.

**Responses**

Response 1 (ALL, CLASS, NAME, spoolid):

This response format is displayed for an ALL, CLASS, NAME, or *spoolid* request if SHORTDATE is specified or is the user's default date format. The response contains one record for each file that matches the criteria specified on the command. The fields in the response contain the following information:

OWNERID	FILE TYPE	CL	RECS	DATE	TIME	FILENAME	FILETYPE	ORIGINID
*NSS	<i>spid</i> NSS	<i>c</i>	<i>recs</i>	<i>mm/dd</i>	<i>hh:mm:ss</i>	<i>fn</i>	<i>ft</i>	<i>originid</i>

Response 2 (ALL, CLASS, NAME, spoolid):

This response format is displayed for an ALL, CLASS, NAME, or *spoolidrequest* if FULLDATE is specified or is the user's default date format. The response contains one record for each file that matches the criteria specified on the command. The fields in the response contain the following information:

OWNERID	FILE	TYPE	CL	RECS	DATE	TIME	FILENAME	FILETYPE	ORIGINID
*NSS	<i>spid</i>	NSS	<i>c</i>	<i>recs</i>	<i>mm/dd/yyyy</i>	<i>hh:mm:ss</i>	<i>fn</i>	<i>ft</i>	<i>originid</i>

Response 3 (ALL, CLASS, NAME, spoolid):

This response format is displayed for an ALL, CLASS, NAME, or *spoolidrequest* if ISODATE is specified or is the user's default date format. The response contains one record for each file that matches the criteria specified on the command. The fields in the response contain the following information:

OWNERID	FILE	TYPE	CL	RECS	DATE	TIME	FILENAME	FILETYPE	ORIGINID
*NSS	<i>spid</i>	NSS	<i>c</i>	<i>recs</i>	<i>yyyy-mm-dd</i>	<i>hh:mm:ss</i>	<i>fn</i>	<i>ft</i>	<i>originid</i>

**OWNERID**

identifies the owner of the file. The designation \*NSS indicates that the file is owned by the system. \*NSS is not a user ID; it is the name of the queue on which the file resides.

**FILE**

identifies the spool file number.

**TYPE**

identifies the type of system data file. All files containing NSSs or saved segments are type NSS.

**Note:** Do not confuse this with the file type of the system data file.

**CL**

identifies the file class. For the list of these classes, see "Response 7 (MAP)" under ["Responses"](#) on page 1042.

**RECS**

specifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where *nnn* is the number of lines in thousands rounded to the nearest thousand. If the number is greater than 999499, the number is shown as *nnnM*, where *nnn* is the number of lines in millions rounded to the nearest million.

**DATE**

identifies the date the NSS or saved segment was defined. The format of the date is determined by the SHORTDATE, FULLDATE, or ISODATE operand. If no date format operand is specified, the user's default date format setting is used.

**TIME**

identifies the time the NSS or saved segment was defined, in hours, minutes, and seconds.

**FILENAME**

identifies the file name of the system data file that contains the NSS or saved segment.

**FILETYPE**

identifies the file type of the system data file. This is NSS for a named saved system, DCSSG for a discontinuous saved segment (DCSS) above 2047 MB, and DCSS for any other type of saved segment.

**ORIGINID**

identifies the user ID of the user who saved the NSS or saved segment (for class A, R, or P files). If the file is a skeleton (class S), this is the user who defined the NSS or saved segment.

Response 4 (ATTRIBUTES):

FILE	FILENAME	FILETYPE	CL	ATTRIBUTES
<i>spid</i>	<i>fn</i>	<i>ft</i>	<i>c</i>	<i>attr</i>

This response format is displayed for an ATTRIBUTES request. The response contains one record for each file that matches the criteria specified on the command. The fields in the response contain the following information:

## QUERY NSS

### FILE

identifies the spool file number.

### FILENAME

identifies the file name of the system data file that contains the NSS or saved segment.

### FILETYPE

identifies the file type of the system data file:

<u>Type</u>	<u>Meaning</u>
NSS	Named saved system
DCSS	DCSS below 2047 MB
DCSSG	DCSS above 2047 MB
DCSS-S	Segment space
DCSS-M	Member segment
CPNSS	CP system service NSS
CPDCSS	CP system service DCSS

### CL

identifies the file class. For the list of these classes, see "Response 7 (MAP)" under "[Responses](#)" on [page 1042](#).

### ATTRIBUTES

identifies the set of additional attributes that pertain to this file. Items in this list are delimited by a blank. Possible values are:

<u>Value</u>	<u>Meaning</u>
LOADNSHR	This attribute pertains only to DCSSs and segment spaces. It means that this saved segment may be loaded nonshared (using DIAGNOSE code X'64') by any user. Users do not need NAMESAVE directory privilege to obtain an exclusive write copy of this saved segment.
NONE	There are no additional attributes for this NSS or saved segment. Enter the QUERY NSS command with the MAP option to get more information about this file.

Response 5 (COUNT):

```
FILES: {nnnn} NSS
       { NO }
```

This response format is displayed for a COUNT request, or for any other version of the command (except QUERY NSS USERS *fn*) if no files match the selection criteria.

### Value

#### Meaning

### **nnnn**

is the number of NSS and saved segment files that match the selection criteria.

### **NO**

No files match the selection criteria.

Response 6 (ENVIRONMENT):

```
FILE FILENAME FILETYPE CL MINSIZE MACHMODE
spid fn          ft      c  minsize mm1(,mm2,...)
```

This response format is displayed for an ENVIRONMENT request and supplies information about NSS files only. The response contains one record for each file that matches the criteria specified on the command. The fields in the response contain the following information:

**FILE**

identifies the spool file number.

**FILENAME**

identifies the file name of the system data file that contains the NSS.

**FILETYPE**

identifies the file type of the system data file. This is either NSS or CPNSS.

**CL**

identifies the file class. For the list of these classes, see "Response 7 (MAP)" under ["Responses"](#) on page 1042.

**MINSIZE**

specifies the minimum storage size of the virtual machine into which the NSS can be loaded. Storage is represented in multiples as follows:

<u>Representation</u>	<u>Multiples</u>
<i>nnnnnnnK</i>	Decimal kilobytes
<i>000nnnnM</i>	Decimal megabytes

**MACHMODE**

identifies the set of virtual machine modes in which this NSS can be IPLed:

<u>Value</u>	<u>Meaning</u>
<i>mm1,mm2,...</i>	This is a list of one or more virtual machine modes, separated by commas with no intervening blanks. Each entry in this list is either ESA, XC, or Z, corresponding to ESA/390, ESA/XC, or z/Architecture respectively. An NSS with more than one mode listed is called a multimode NSS.
<i>—</i>	This NSS is a skeleton and the MACHMODE operand was not specified on the DEFSYS command. The virtual machine mode in which this NSS can be IPLed is that of the virtual machine that issues the SAVESYS command.

Response 7 (MAP):

This response is displayed for a MAP request. The response contains one record for each file that matches the criteria specified on the command; more than one page range may be displayed.

Also, for a QUERY NSS NAME *fn* MAP request:

- If *fn* is a member of a segment space, the response contains one record for the member, followed by one record for the space.
- If *fn* is a segment space, the response contains one record for the space, followed by one record for each member of the space.

For an NSS, a segment space, a member saved segment, or a DCSS below 2047 MB, the response format is:

```
FILE FILENAME FILETYPE MINSIZE BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
spid fn ft minsize nnnnn nnnnn type c nnnnn parmregs vmgroup
```

For a DCSS above 2047 MB, the response format is:

```
FILE FILENAME FILETYPE BEGPAG ENDPAG TYPE CL #USERS
spid fn DCSSG nnnnnnnnnnnnnn nnnnnnnnnnnnnn type c nnnnn
```

**Note:** The format for NSSs and saved segments below 2047 MB will be displayed first in the response, followed by the format for DCSSs above 2047 MB.

## QUERY NSS

The fields in the response contain the following information:

### FILE

identifies the spool file number.

### FILENAME

identifies the file name of the system data file.

### FILETYPE

identifies the file type of the system data file:

<u>Type</u>	<u>Meaning</u>
NSS	Named saved system
DCSS	DCSS below 2047 MB
DCSSG	DCSS above 2047 MB
DCSS-S	Segment space
DCSS-M	Member segment
CPNSS	CP system service NSS
CPDCSS	CP system service DCSS

### MINSIZE

for NSS files only, specifies the minimum storage size of the virtual machine into which the NSS can be loaded. This field does not apply to a saved segment, for which N/A is displayed.

Storage is represented in multiples as follows:

<u>Representation</u>	<u>Multiples</u>
<i>nnnnnnnK</i>	Decimal kilobytes
<i>000nnnnM</i>	Decimal megabytes

### BEGPAG

specifies the beginning page number of a page range of the NSS or saved segment. For a segment space, this is the beginning page number of the entire segment space.

### ENDPAG

specifies the ending page number of a page range of the NSS or saved segment. For a segment space, this is the ending page number of the entire segment space.

### TYPE

identifies the page descriptor code of a page range of the NSS or saved segment. It indicates the type of virtual machine access permitted to pages in the range. See the DEFSEG or DEFSYS command for a more detailed discussion of page descriptor codes. Valid codes are:

#### Code

#### Meaning

#### EW

Exclusive read/write access.

#### EN

Exclusive read/write access, no data saved.

#### ER

Exclusive read-only access.

#### SW

Shared read/write access.

#### SN

Shared read/write access, no data saved.

**SR**  
Shared read-only access.

**SC**  
CP writable pages, shared read-only access by virtual machine, no data saved.

For a segment space, the access type might not apply to the entire range, so a dash (-) is displayed.

**CL**  
indicates the current class (state) of the system data file. The possible file classes are:

**Class**  
**Meaning**

**A**  
Available state. This means the system data file has been defined and saved. To determine whether the NSS or saved segment is in use, examine the #USERS field in the response.

**P**  
Pending purge state. This means the PURGE NSS command (or DIAGNOSE code X'64') has been issued for the name of this NSS or saved segment but virtual machines are still accessing it. No new users can access this NSS or saved segment. The file will be purged when the last virtual machine releases the NSS or saved segment, or during the next system IPL or RESTART.

**R**  
Restricted access available state. This means the system data file has been defined (with the RSTD option) and saved. Access to a restricted NSS or saved segment requires a NAMESAVE directory statement. To determine whether the NSS or saved segment is in use, examine the #USERS field in the response.

**S**  
Skeleton state. This means the file has been defined by a DEFSEG or DEFSYS command, and the SAVESEG or SAVESYS command can now be run to complete this system data file.

**#USERS**  
specifies the number of users attached to the NSS or saved segment.

**Note:** The number of users pertains to that file name only.

**PARMREGS**  
for NSS files only, identifies the registers in which parameters are passed to the virtual machine at IPL. Possible values are:

<u>Value</u>	<u>Meaning</u>
<i>m-n</i>	A range of registers, where <i>m</i> and <i>n</i> are decimal integers between 00 and 15 (inclusive) and <i>m</i> is less than or equal to <i>n</i> .
NONE	PARMREGS=NONE was specified on the DEFSYS command.
OMITTED	The PARMREGS option was not specified on the DEFSYS command.

This field does not apply to a saved segment, for which N/A is displayed.

**VMGROUP**  
for NSS files only, specifies whether the NSS is part of a virtual machine group. Possible values are:

**Value**  
**Meaning**

**YES**  
A user who IPLs the NSS becomes part of the virtual machine group, identified by the name of the NSS.

**NO**  
The VMGROUP option was not specified on the DEFSYS command.

This field does not apply to a saved segment, for which N/A is displayed.

## QUERY NSS

Response 8 (USERS):

```
FILE FILENAME FILETYPE CLASS
  spid fn      ft      cl
  userid1 userid2 userid3 ...
```

This response format is displayed for a USERS request. More than one response may be displayed:

- If *fn* is an NSS or DCSS, one response is displayed.
- If *fn* is a member of a segment space, one response is displayed for the member saved segment, followed by one response for the segment space.
- If *fn* is a segment space, one response is displayed for the segment space, followed by one response for each member saved segment.

The fields in the response contain the following information:

### FILE

identifies the spool file number.

### FILENAME

identifies the file name of the system data file.

### FILETYPE

identifies the file type of the system data file:

<u>Type</u>	<u>Meaning</u>
NSS	Named saved system
DCSS	DCSS below 2047 MB
DCSSG	DCSS above 2047 MB
DCSS-S	Segment space
DCSS-M	Member segment
CPNSS	CP system service NSS
CPDCSS	CP system service DCSS

### CLASS

indicates the file class. For the list of these file classes, see "Response 7 (MAP)" under ["Responses"](#) on page 1042.

### *userid1 userid2 userid3 ...*

is a list of the user IDs of the users who are using the NSS or saved segment. In the display for a segment space and its members, the space itself and some members may not currently be loaded by any user. In that case, the list of users is replaced by the word NONE.

**Note:** A user ID of SYSTEM in the response means that part of the CP system is using the DCSS or NSS. There can be multiple parts of the CP system using a DCSS or NSS at the same time. Therefore, the response to QUERY NSS USERS could have SYSTEM listed more than once.

Response 9 (USERS):

```
fn NOT IN USE OR AN INVALID NSS/DCSS NAME
```

This response is displayed for a USERS request if the specified name is nonexistent or not in use.

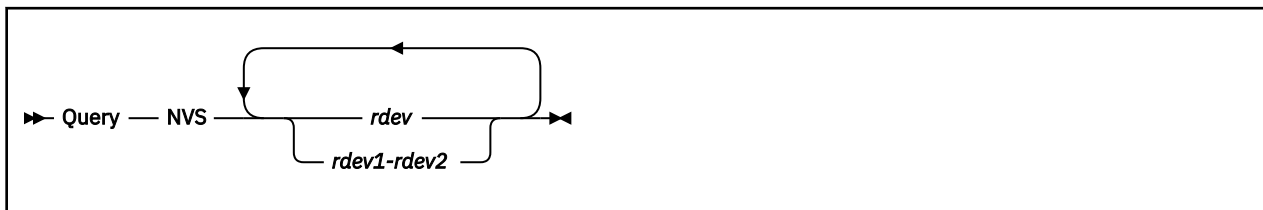
## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP413E Command processing cannot complete due to a spool paging error.



- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP849E A user ID must be specified with the spoolid
- HCP1558E Command processing cannot complete.

## QUERY NVS



### Authorization

Privilege Class: B

### Purpose

Use QUERY NVS to display the status of the nonvolatile storage (NVS) associated with a DASD storage subsystem.

### Operands

**rdev**

**rdev1-rdev2**

is the real device number, a list of real device numbers, or a range of real device numbers of devices in the subsystem being queried.

### Usage Notes

1. QUERY NVS is only valid for cache storage subsystems that have nonvolatile storage. An error message is generated if the specified subsystem does not have nonvolatile storage.
2. Caching for RAID (Redundant Array of Independent Disks) technology DASD such as an IBM DASD subsystem is always available. It cannot be modified by the user.

### Responses

Response 1:

If the amount of configured and pinned bytes are able to be determined, the following response is issued.

```

rdev Nonvolatile storage {is available.
                        {is unavailable.
                        {has failed.
                        {is disabled for maintenance.
                        {is being destaged.
                        {destaging has failed.

ccccccccK Bytes are configured.
ppppppppK Bytes are pinned.
  
```

This response is repeated for each *unique subsystem* identified by the devices specified on the command line.

**rdev**

is the real device number of a device in the subsystem.

**cccccccc**

is the configured size of the nonvolatile storage in kilobytes (1024-byte blocks).

**pppppppp**

is the size in kilobytes (1024-byte blocks) of nonvolatile storage that is not available because of exception conditions that prevent successful destaging of data to the backing storage device.

Response 2:

If the amount of configured and pinned bytes cannot be determined, the following response is issued.

```

rdev Nonvolatile storage {is available.
                        {is unavailable.
                        {has failed.
                        {is disabled for maintenance.
                        {is being destaged.
                        {destaging has failed.

```

```

Bytes configured cannot be determined.
Bytes pinned cannot be determined.

```

This response is repeated for each *unique subsystem* identified by the devices specified on the command line.

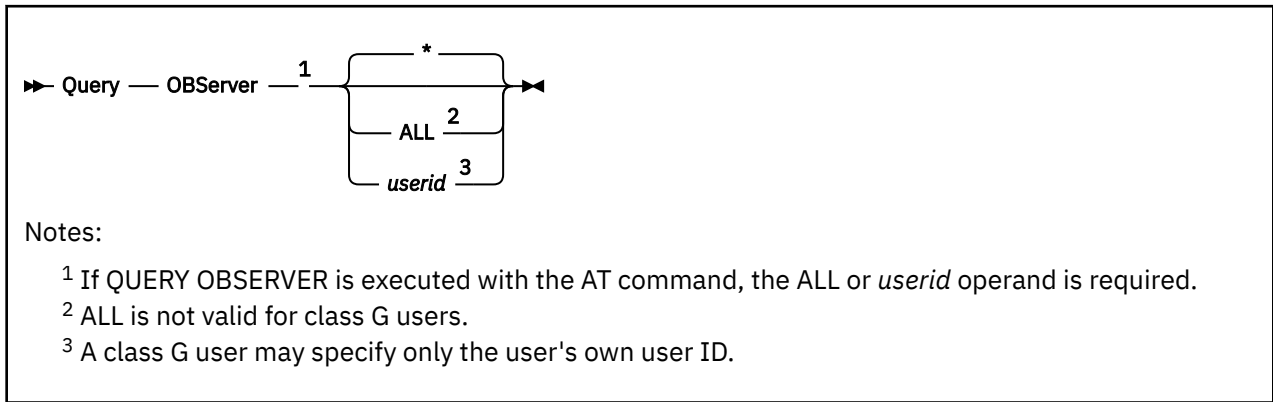
**rdev**

is the real device number of a device in the subsystem.

## Messages

- HCP006E Invalid device type - {rdev/vdev/ldev}
- HCP009E Invalid range - range
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP046E type rdev offline
- HCP319E Error processing subsystem command - cmd for DASD rdev
- HCP332E Invalid control unit type - rdev

## QUERY OBSERVER



### Authorization

Privilege Class: A, B, C, G

### Purpose

Use QUERY OBSERVER to determine the observer setting for a virtual machine. CP displays the identity and status of the observer. If you are a class G user, you can display the observer setting for your own virtual machine. If you are a class A, B, or C user, you can display the observer setting for any virtual machine or for all local virtual machines that have observers defined.

### Operands

**\***

displays the observer setting for the user who issues the command. This is the default.

**ALL**

displays the observer setting for all local users that have observers defined.

***userid***

identifies the user whose observer setting is to be displayed.

### Usage Notes

1. The initial setting for your observer is determined by the CONSOLE statement in your user directory entry.
2. A virtual machine cannot have both an observer and a secondary user defined by any combination of the SET OBSERVER and SET SECUSER commands and the CONSOLE directory statement.
3. In an SSI cluster, the observer and the target (observed) user can be logged on to different member systems.

If either user is a multiconfiguration virtual machine, the observer will function in that capacity only when it is local (logged on to the same member as the observed user). If the observed user is a multiconfiguration virtual machine, the observer must be logged on to the same member as the instance of the observed user for which the observer setting was established:

- If the setting was established by a CONSOLE statement in the identity entry in the observed user's virtual machine definition, a potential observer relationship exists for every member instance of the observed user.

- If the setting was established by a CONSOLE statement in a subconfiguration entry in the observed user's virtual machine definition, a potential observer relationship exists for that member instance of the observed user.
- If the setting was established by a SET OBSERVER command, a potential observer relationship exists for the member on which the SET OBSERVER command was issued.

## Responses

Responses 1 through 5 are returned if you issue the QUERY OBSERVER command without any additional operands.

Response 1:

```
OBSERVER userid IS DISCONNECTED
```

is returned if an observer is defined for your virtual machine (on the CONSOLE statement in your virtual machine definition or on a SET OBSERVER command), and that user is disconnected.

If the observer reconnects, it can provide the observer function for your virtual machine.

Response 2:

```
OBSERVER userid IS NOT LOGGED ON
```

is returned if an observer is defined for your virtual machine (on the CONSOLE statement in your virtual machine definition or on a SET OBSERVER command), and that user is not logged on.

If the observer logs on, it can provide the observer function for your virtual machine.

Response 3:

```
OBSERVER userid IS LOGGED ON
```

is returned if an observer is defined for your virtual machine (on the CONSOLE statement in your virtual machine definition or on a SET OBSERVER command), and that user is logged on.

Response 4:

```
OBSERVER IS UNDEFINED
```

is returned if an observer is not defined for your virtual machine.

Response 5:

```
OBSERVER IS NOT LOGGED ON LOCALLY
```

is returned if all of the following conditions are true:

- This system is a member of an SSI cluster.
- You, the observed user, are defined as a multiconfiguration virtual machine.
- An observer is defined for your virtual machine on this member of the cluster (on a CONSOLE statement in the subconfiguration entry for this member in your virtual machine definition, or on a SET OBSERVER command issued on this member).
- That user is not logged on to this member of the cluster. (The user might be logged on to another member.)

Response 6:

```
FAILED MAC CHECK
```

is returned if security label checking by an external security manager (ESM) is active and the observed/observer user relationship fails the mandatory access control checks.

## QUERY OBSERVER

**Response 7:** If you issue the QUERY OBSERVER command with an additional option and the observer is disconnected, CP responds with a display that looks like this:

Userid	Observer	Status
MAINT	DAVID	disconnected

The Status column in the display might also contain "not logged on", "logged on", "not defined", "not logged on locally", or "failed MAC check".

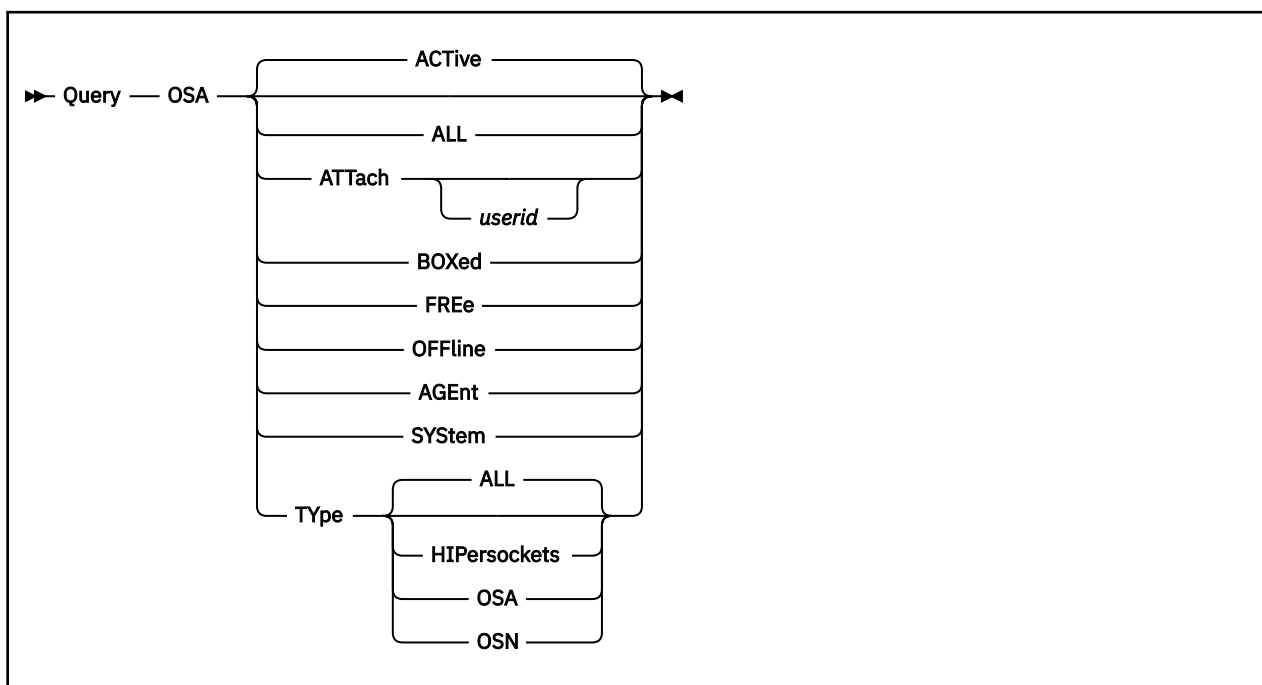
**Response 8:** If a class A, B, or C user issues QUERY OBSERVER ALL, CP displays the current observer settings for all users that are currently logged on or disconnected. If no currently logged on users have an observer defined, CP issues the following response:

No observers defined or active.

### Messages

- HCP003E Invalid option - *option*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP045E *userid* not logged on
- HCP1149E Error communicating with another SSI member. It cannot be determined if *command* executed remotely.

## QUERY OSA



### Authorization

Privilege Class: B

### Purpose

Use QUERY OSA to display the status of real Open Systems Adapter devices.

### Operands

#### ACTive

indicates active Open Systems Adapter devices. An active device is one that is in use by a user or the system and is specified as neither FREE nor OFFLINE nor SYSTEM.

#### ALL

indicates all Open Systems Adapter devices attached to the system. The status is displayed in the order of ACTIVE, SYSTEM, FREE, OFFLINE, and AGENT. You receive the same response as you would receive if you entered all of the following commands:

```

query osa active
query osa system
query osa free
query osa offline
query osa agent
  
```

#### ATTach *userid*

indicates the OSA devices that are dedicated to a user on the system. If you specify *userid*, only OSA devices attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all OSA devices dedicated to any user on the system. Attached devices are always active.

#### BOXed

indicates the OSA devices that are unavailable (usually because of a hardware condition). A boxed device can also be active, free, or offline.

## QUERY OSA

### **FREe**

indicates all free OSA devices. A free OSA device is one that is not currently in use by a user or the system and is specified as neither ACTIVE nor OFFLINE.

### **OFFline**

indicates all offline OSA devices. An offline OSA device is one that is specified as neither ACTIVE nor FREE.

### **AGEnt**

indicates all Open Systems Adapter Agent devices. OSA Agent devices are used to control the OSA configuration.

### **SYStem**

indicates all OSA devices reserved for system use. An OSA device that is reserved for system use is associated with an OSA adapter that is in use on a shared port group.

### **TYpe**

indicates the type of OSA devices to be displayed. For each device, the display includes the type of device, the channel-path identifier (CHPID), and the type of channel path.

### **ALL**

indicates that all types of OSA devices are to be displayed. This is the default value.

### **HIPersockets**

indicates that HiperSockets devices are to be displayed. HiperSockets devices are used for communication between logical partitions and/or virtual machines, and are associated with internal-queued-direct-communication (IQD) channel paths. The displayed devices will include device types of HIPER as well as HIPER-BRDG. HIPER-BRDG indicates the EXTERNAL\_BRIDGED channel parameter was configured for the channel path.

### **OSA**

indicates that OSA devices are to be displayed. OSA devices are used for communication through a network, and are associated with Open Systems Adapter (OSA) channel paths, OSA-Express (OSE) channel paths, or OSA Direct Express (OSD) channel paths.

### **OSN**

indicates that OSN devices are to be displayed. OSA NCP (OSN) devices are used by the Communication Controller for Linux (CCL) when running the Network Control Program (NCP) in a logical partition or virtual machine. OSN devices are associated with OSN channel paths and communicate with one or more emulated 3745 device types on the same OSN channel path.

## Usage Notes

1. A VIRTUAL version of this command, QUERY VIRTUAL OSA, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL OSA” on page 1316](#).

## Responses

If the system is using an OSA device that is in an intervention-required state and you enter QUERY OSA, you see a response like this:

```
OSA 0300 has intervention required.
```

0300 is the real device number of the OSA device that needs the operator's intervention.

**Note:** This response will be preceded by one of the following.

Response 1:

If you enter QUERY OSA, QUERY OSA ACTIVE, or QUERY OSA ATTACHED and there is an OSA device attached to the system, you see something like this:

```
q osa active
OSA 0480 ATTACHED TO PVM      0480 DEVTYPE HIPER      CHPID 80 IQD
```



```
OSA 0481 ATTACHED TO PVM      0481 DEVTYPE HIPER      CHPID 80 IQD
OSA 0482 ATTACHED TO PVM      0482 DEVTYPE HIPER      CHPID 80 IQD
OSA 0888 ATTACHED TO RSCS     0888 DEVTYPE OSA        CHPID A0 OSD
OSA 0889 ATTACHED TO RSCS     0889 DEVTYPE OSA        CHPID A0 OSD
OSA 088A ATTACHED TO RSCS     088A DEVTYPE OSA        CHPID A0 OSD
OSA 1700 ATTACHED TO VTAM     0500 DEVTYPE OSN        CHPID E0 OSN
```

Ready;

The first number is the real device number of the OSA device. The ATTACHED operand might also read BOX/ATTC, which means that although the device is attached and dedicated to a user, it is unavailable. PVM, RSCS, TCPIP, and VTAM are user IDs to which the OSA devices are dedicated. The next number is the virtual device number used by a virtual machine to refer to the specified OSA device. Following the virtual device number is the type of the device, the channel path identifier (CHPID) associated with the device, and the channel-path type.

Response 2:

If you enter a QUERY OSA FREE command and there is a free OSA device attached to the system, you see something like this:

```
q osa free
OSA 0487 FREE      , OSA 04A1 FREE      , OSA 04A2 FREE      , OSA 04A3 FREE
OSA 04A4 FREE      , OSA 04A5 FREE      , OSA 04A9 FREE      , OSA 04AA FREE
OSA 04AB FREE      , OSA 04BC FREE      , OSA 04BD FREE      , OSA 04BE FREE
OSA 04BF FREE
Ready;
```

This response shows the real device number for each free OSA device.

The response for a QUERY OSA OFFLINE or QUERY OSA BOXED command is in the same format as the response for a QUERY OSA FREE command.

Response 3:

When you enter a QUERY OSA AGENT command and there is an OSA AGENT attached to the system, you see something like this:

```
q osa agent
OSA 0280 is an OSA Agent
```

Response 4:

When you enter a QUERY OSA ALL command, CP displays the same responses as it would display if you had entered the QUERY OSA ACTIVE, QUERY OSA SYSTEM, QUERY OSA FREE, QUERY OSA OFFLINE, and QUERY OSA AGENT commands.

Response 5:

If you enter QUERY OSA TYPE ALL, you will see something like this:

```
q osa type all
OSA 0480 ATTACHED TO TCPIP     0100 DEVTYPE HIPER      CHPID 80 IQD
OSA 0481 BOX/ATTC TO TCPIP     0101 DEVTYPE HIPER      CHPID 80 IQD
OSA 0482 ATTACHED TO TCPIP     0102 DEVTYPE HIPER      CHPID 80 IQD
OSA 7240 SYSTEM
OSA 7241 SYSTEM
OSA 7242 SYSTEM
OSA 0484 FREE                  DEVTYPE OSA             CHPID C0 OSE
OSA 0485 FREE                  DEVTYPE OSA             CHPID C0 OSE
OSA 0486 FREE                  DEVTYPE OSA             CHPID D0 OSA
OSA 0487 FREE                  DEVTYPE OSA             CHPID D0 OSA
OSA 04A1 OFFLINE              DEVTYPE OSA             CHPID D1 OSD
OSA 04A2 OFFLINE              DEVTYPE OSA             CHPID D1 OSD
OSA 04A3 OFFLINE              DEVTYPE OSA             CHPID D1 OSD
OSA 04A4 OFFLINE              DEVTYPE OSA             CHPID D1 OSD
OSA 04A5 OFFLINE              DEVTYPE OSA             CHPID D1 OSD
OSA 04A9 OFFLINE              DEVTYPE OSA             CHPID D1 OSD
OSA 04AA OFFLINE              DEVTYPE OSA             CHPID D1 OSD
OSA 04AB BOXED                DEVTYPE OSA             CHPID D1 OSD
OSA 04BC FREE                 DEVTYPE OSA             CHPID D1 OSD
OSA 04BD FREE                 DEVTYPE OSA             CHPID D1 OSD
OSA 048E FREE                 DEVTYPE OSA             CHPID D1 OSD
```

## QUERY OSA

```
OSA 04BF FREE          DEVTYP OSA          CHPID D1 OSD
OSA 0700 FREE          DEVTYP OSN          CHPID E0 OSN
OSA 0701 FREE          DEVTYP OSN          CHPID E0 OSN

OSA 1818 OFFLINE      DEVTYP INMN         CHPID F1 OSM
OSA 1819 OFFLINE      DEVTYP INMN         CHPID F1 OSM
OSA 181A OFFLINE      DEVTYP INMN         CHPID F1 OSM
OSA 3F00 OFFLINE      DEVTYP IEDN         CHPID F6 OSX
OSA 3F01 OFFLINE      DEVTYP IEDN         CHPID F6 OSX
OSA 3F02 OFFLINE      DEVTYP IEDN         CHPID F6 OSX
Ready;
```

This response shows the real device number of the OSA device with its current status, followed by the device type, the associated channel-path identifier, and the channel-path type. The following device status may be returned:

ATTACHED TO userID vdev	Attached and dedicated to the specified userID as virtual device vdev
BOX/ATTC TO userID vdev	Attached and dedicated to the specified userID as virtual device vdev, but is not available
BOXED	Device is not available
OFFLINE	Device is OFFLINE
FREE	Device is ONLINE and available to use
SYSTEM	Device is reserved for system use

Response 6:

If you enter QUERY OSA TYPE HIP, you will see something like this:

```
q osa type hip
OSA 0480 ATTACHED TO PVM          0480 DEVTYP HIPER          CHPID 80 IQD
OSA 0481 ATTACHED TO PVM          0481 DEVTYP HIPER          CHPID 80 IQD
OSA FD00 ATTACHED TO DTCVSW1     FD00 DEVTYP HIPER-BRDG     CHPID FD IQD
OSA FD01 ATTACHED TO DTCVSW1     FD01 DEVTYP HIPER-BRDG     CHPID FD IQD
OSA FD02 ATTACHED TO DTCVSW1     FD02 DEVTYP HIPER-BRDG     CHPID FD IQD
Ready;
```

This response shows the real device number of each HiperSockets device with its status, followed by the device type, the associated channel-path identifier, and the channel-path type. A device type of HIPER-BRDG indicates the EXTERNAL\_BRIDGED channel parameter was configured for the channel path.

Response 7:

If you specify the *type* parameter and CP cannot find an OSA device of the type you specify, you see a message like this:

```
q osa type osn
An OSA was not found for the specified type.
Ready;
```

Response 8:

If CP cannot find an OSA device for the query you specify, you see a message like this:

```
q osa boxed
A BOXED OSA was not found.
Ready;
```

In this response, BOXED may be replaced with *active*, *attached*, *offline*, *system*, or *free*, depending on what you specified on the command. The BOXED OSA would be replaced by OSA AGENT in this response if agent was specified on the command.

## Messages

- HCP003E Invalid option - *option*

- HCP006E Invalid device type - *rdev*
- HCP007E Invalid userid - *userid*
- HCP263E Too many operands were supplied for this command.

## QUERY PAGING

► Query — PAGING ◄

### Authorization

Privilege Class: B, E

### Purpose

Use the QUERY PAGING command to display settings for the paging subsystem.

### Usage Notes

1. Use the PAGING statement in the system configuration file to define the paging subsystem's settings at IPL.  
For compatibility, the Alias and HPF settings can also be specified on the FEATURES statement in the system configuration file, using the PAGING\_ALIAS and PAGING\_HPF operands. However, it is recommended that you consolidate your paging options on the PAGING statement.  
You can use the SET PAGING command after IPL to alter the paging subsystem's settings.
2. HyperPAV aliases allow concurrent I/O requests to the same physical volume by using alias devices. Aliases do not have associated storage and can be bound dynamically to base volumes. Using HyperPAV can increase throughput by reducing queuing at the device level.
3. High Performance FICON improves paging efficiency through its support for transport mode channel program format. With transport mode, less overhead occurs between the channel subsystem and the FICON adapter than with traditional command-mode I/O. This results in higher I/O rates and less CPU overhead.
4. The warning value is also used by the SET STORAGE command to make sure that a storage removal does not cause paging volumes to exceed this level of allocation. For more information, see [“SET STORAGE”](#) on page 1722.

### Responses

Response 1:

```
PAGING_ALIAS: {ON}          PAGING_HPF= {ON}          WARNING: nnn%
               {OFF}         {OFF}
```

#### PAGING\_ALIAS

indicates the state of the HyperPAV alias feature of the paging subsystem.

#### PAGING\_HPF

indicates the state of the HPF feature of the paging subsystem.

##### ON

the feature is active

##### OFF

the feature is not active

#### Warning *nnn%*

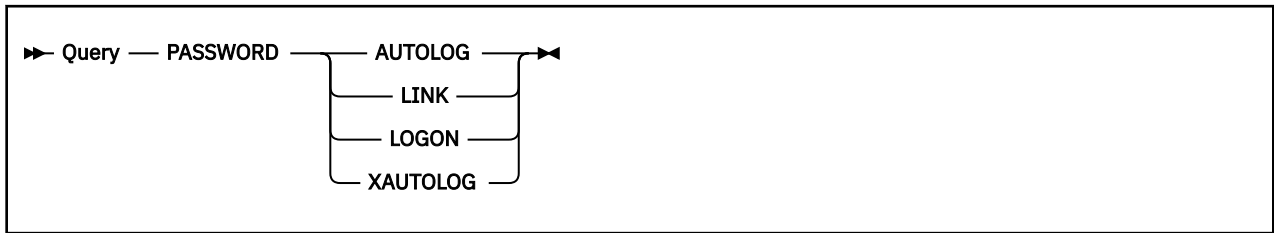
is the allocation warning level for paging volumes, as specified on a PAGING system configuration statement or a previous SET PAGING command. The *nnn* value is a decimal integer in the range of 0 to 100. If no WARNING value has been specified, this value is 90%. If the number of allocated paging slots reaches this percentage of total slots on all paging volumes, message HCP401I is issued

to alert the system operator. This value also serves as the default MAXPAGEFULL setting for a SET STORAGE command to ensure a storage removal does not cause paging volumes to exceed this level of allocation. See [“SET STORAGE” on page 1722](#) for more information.

## Messages

- HCP003E Invalid option - *{option | command contains extra option(s) starting with option}*

## QUERY PASSWORD



### Authorization

Privilege Class: G

### Purpose

Use QUERY PASSWORD to determine whether you can include a password on the command line for the AUTOLOG, LINK, LOGON, or XAUTOLOG command.

### Operands

#### AUTOLOG

displays whether you must include a password on the AUTOLOG command line.

#### LINK

displays whether you can include a password on the LINK command line.

#### LOGON

displays whether you can include a password on the LOGON command line.

#### XAUTOLOG

displays whether you can include a password on the XAUTOLOG command line.

### Usage Notes

1. For security reasons, the password suppression facility permits your installation to require that issuers of commands that require passwords do not include the password in the command line. Instead, after the user enters the command without the password, the system prompts the user to enter the password, and the input field is masked.
2. See [“SET PASSWORD” on page 1598](#) for an explanation of how to set password suppression for the AUTOLOG, LINK, LOGON, and XAUTOLOG commands.
3. If you enter QUERY PASSWORD using DIAGNOSE code X'08', you receive one of the following return codes in your R<sub>y</sub> register:

#### RC

##### Meaning

0

Indicates that you may (must, for AUTOLOG) enter a password on the command line.

4

Indicates that you cannot enter a password on the command line.

8

Error—the operands following on the command line are not valid.

4. System programmers responsible for z/VM security should examine the usage notes for the PASSWORDS\_ON\_CMDs operand of the FEATURES system configuration statement. See the [FEATURES Statement in z/VM: CP Planning and Administration](#).

## Responses

Response 1:

```
{ AUTOLOG } PASSWORD MUST BE ENTERED SEPARATELY  
{ LINK }  
{ LOGON }  
{ XAUTOLOG }
```

indicates that you *cannot* enter the access mode password on the command line.

Response 2:

```
{ LINK } PASSWORD MAY BE INCLUDED ON THE COMMAND LINE  
{ LOGON }  
{ XAUTOLOG }
```

indicates that you *can* enter the access mode password on the command line.

Response 3:

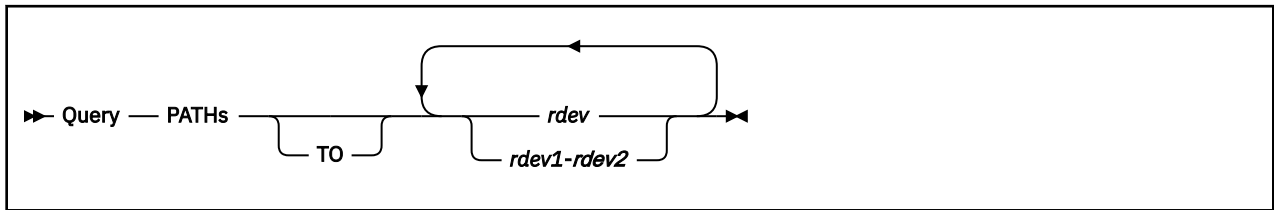
```
{ AUTOLOG } PASSWORD MUST BE INCLUDED ON THE COMMAND LINE
```

indicates that you *must* enter the access mode password on the command line.

## Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid

## QUERY PATHS



### Authorization

Privilege Class: B, E

### Purpose

Use QUERY PATHS to display:

- All paths installed to a specific device or range of devices
- Installed path status

### Operands

*rdev*

*rdev1-rdev2*

**TO** *rdev*

**TO** *rdev1-rdev2*

is the real device number or range of real device numbers whose path allocation information you want. Each *rdev* in the active configuration can be entered as a 4-digit hexadecimal number between X'0000' and X'FFFF'. For a device in the active or standby configuration, the device number can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured. You can specify a single device address, a range of device addresses, or any combination thereof. A range of devices cannot span subchannel sets.

### Usage Notes

1. The standard response lines for QUERY PATHS are shown in the following example. They are called standard because they will always be displayed for a device that has installed paths.

```
Device 0291, Status ONLINE
CHPIDs to Device 0291 (PIM) : 19 1B 2E 2F
Physically Available (PAM) : + + + +
Online (LPM) : + + + +

Legend      + Yes - No
```

2. The query response reflects the current hardware information for paths that are physically installed and physically available. The logical path information you see represents a snapshot of what CP knows based on information in the RDEV.
3. For information about the devices associated with a specific channel path, see [“QUERY CHPID” on page 791](#).
4. Error messages produced for this command may vary depending on how the device was specified on the command line. When a single device is listed on the command line, an error will be reported if there is no query information available for that single device. For example:

```
query paths to 102
HCPQPA0040E Device 0102 does not exist
Ready(00040);
```



When a range of devices is listed on the command line, only devices that have query information available are shown. Errors messages are not issued for individual devices within the range. For example:

```
query paths to 100-105
Device 0105, Status OFFLINE
CHPIDs to Device 0105 (PIM) : 19 1B 2E 2F
  Physically Available (PAM) : + + + +
  Online (LPM) : + + + +

Legend + Yes - No
Ready;
```

If query information is not available for at least one device in a range, then a generic error message will be issued. For example:

```
query paths to 100-102
HCPQPA1003E 0100-0102 was not found.
Ready(01003);
```

## Responses

Response 1 (Generic):

The following is the full, generic response to the QUERY PATHS command. Normally, you will see only pieces of this response, as CP displays only the lines that have data. For example, if the device does not have any offline channel paths, CP would not display lines 6 through 10. The maximum number of channel paths is 8, as shown in the example. CP displays only the number of channel paths attached to the device. The responses that follow this generic one will show examples of these shortened responses.

```
1 Device rdev, Status status
2 CHPIDs to Device rdev (PIM) : chpid1 ..... chpid8
3 Physically Available (PAM) : +/- ..... +/-
4 Online (LPM) : +/- ..... +/-
5 Preferred : +/- ..... +/-
6 Offline by Authorized User : +/- ..... +/-
7 Offline by ESCON Manager : +/- ..... +/-
8 Out of Path Group by Guest : +/- ..... +/-
9 Offline by Control Unit : +/- ..... +/-
10 Offline, Wrong CU Connection: +/- ..... +/-
11 Transport Mode Supported : +/- ..... +/-
12 Encrypted Link : +/- ..... +/-
13 Authenticated Link : +/- ..... +/-
14 Legend + Yes - No
```

### Line

#### Explanations

1

tells you the real device number of the device and its status. The status field can contain the following:

**BOXED**

device is not available to CP, usually because of a hardware condition

**DISABLE**

device is disabled

**ENABLE**

device is enabled

**OFFLINE**

device is offline

**ONLINE**

device is online

**QUIESCED**

device is offline for maintenance

### **SUBCHOFF**

device is subchannel disabled, see [“VARY SUBCHANNEL”](#) on page 2067.

### **TEST-ONL**

device has been forced online in test mode, see [“VARY \(Real Device\)”](#) on page 2063.

**2**

identifies the real device number and the channel path identifiers (CHPIDs) of all paths physically installed to that device. This is also known as the path installed mask (PIM).

**3**

identifies the paths that are physically available to the device. This is also known as the path available mask (PAM).

**4**

identifies the paths that are logically online to the device. This is also known as the logical path mask (LPM).

**5**

identifies the preferred paths to the device, if applicable.

**6**

identifies the paths that are logically offline because the VARY PATH command was issued by an authorized user.

**7**

identifies the paths that are logically offline because the ESCON configuration program has taken them offline.

**8**

identifies the paths removed from the path group by a guest that is managing the paths to a device.

**9**

identifies the channel paths that are unavailable to the device and locked offline while service is applied to this channel path. Once service on this channel path is completed, a corresponding resume issued by the Service Representative (SR) will cause this channel path to be varied online to the device. If for some reason the resume request did not vary this channel path back online, the FORCE option on the VARY PATH command can be used to override the quiesced path and vary the channel path online to the device. The FORCE option should only be used when the SR indicates that service is complete and a resume was issued but did not vary the channel path online.

**10**

identifies the paths that are logically offline because the wrong control unit connection was detected. This could happen as a result of a cable swap, for example.

**11**

indicates whether CHPIDs are enabled for Transport Mode. Use of High Performance FICON for IBM Z (zHPF) requires channel paths capable of Transport Mode. Moreover, a guest with the CHPIDVIRTUALIZATION ONE directory option can use zHPF only if Transport Mode is supported on all FICON channels in the system, or, in an SSI cluster, on all systems in the guest's assigned relocation domain.

**12**

identifies the paths that are encrypted.

**13**

identifies the paths that are authenticated.

**14**

identifies the legend information.

Response 2 (Path Status):

The following is an example of a query response you may see after an operator has issued the VARY PATH command:

```
Device 0291, Status ONLINE
CHPIDs to Device 0291 (PIM) : 19 1B 2E 2F
Physically Available (PAM) : + + + +
```

```

Online          (LPM) : - - - +
Offline by Authorized User : + + + -
Legend          + Yes - No
    
```

This example shows:

- device 291 is online and there are 4 paths physically installed to the device (19,1B,2E,2F)
- 1 path (2F) is logically online
- 3 paths (19,1B,2E) are logically offline because someone issued the VARY PATH command.

Response 3:

The following is an example of a query response you may see for a device that has never been online (for example, the device may have been listed in the DEVICES statement in the configuration file as NOTINITIALIZED\_at\_jpl):

```

Device 019E, Status SUBCHOFF
CHPIDs to Device 019E (PIM) : 19 1B 2E 2F
Physically Available (PAM) : + + + +
Online          (LPM) : - - - -
Legend          + Yes - No
    
```

This example shows:

- device 19E is in the subchannel disable state and has 4 paths physically installed to the device (19,1B,2E,2F)
- all paths are logically offline

Response 4:

The following is an example of a query response you may see when several paths are offline for different reasons:

```

Device 0192, Status ONLINE
CHPIDs to Device 0192 (PIM) : 11 12 1A 22 30 40
Physically Available (PAM) : + + + + + +
Online          (LPM) : - - + + + -
Offline by Authorized User : + - - - - -
Offline by Control Unit   : - + - - - +
Legend          + Yes - No
    
```

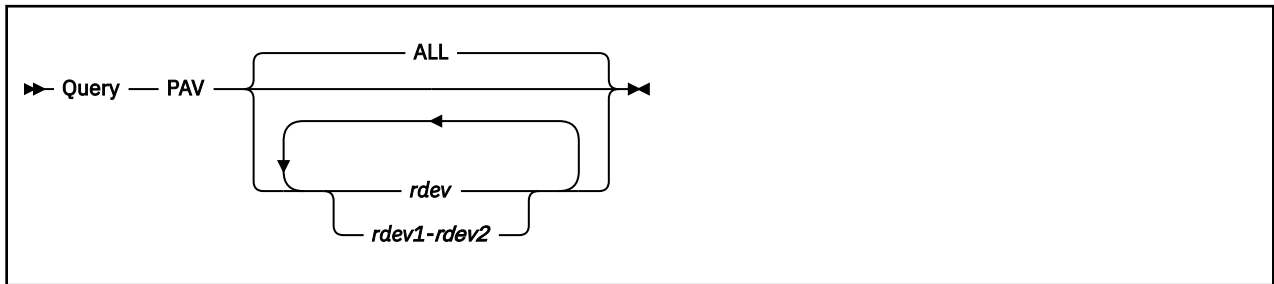
This example shows:

- device 192 is in the online state and has 6 paths physically installed to the device (11,12,1A,22,30,40)
- 3 paths (1A,22,30) are logically online to the device
- 3 paths (11,12,40) are logically offline to the device
- 1 path (11) is logically offline because someone issued the VARY PATH command.
- 2 paths (12,40) are logically offline because someone has issued a command at the control unit in order to service the device

## Messages

- HCP003E Invalid option - {option|command contains extra option(s) starting with option}
- HCP009E Invalid range - rdev-rdev
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device rdev does not exist
- HCP1003E rdev-rdev was not found.
- HCP1150E DASD {rdev/vdev} is not a valid base exposure.
- HCP6706E Invalid RDEV - rdev. RDEV not initialized

## QUERY PAV



### Authorization

Privilege Class: B

### Purpose

Use QUERY PAV to display the following:

- A list of all Parallel Access Volume (PAV) DASD and HyperParallel Access Volume (HyperPAV) DASD. This includes each PAV base device and its associated PAV alias devices, each HyperPAV base device, and each HyperPAV alias device.
- The PAV and/or HyperPAV information for one or more real devices. This includes all base and alias information that is associated with the specified device or devices.

### Operands

#### ALL

tells CP to display a list of all PAV and HyperPAV devices.

#### rdev

#### rdev1-rdev2

is the real device address of the device for which you want PAV or HyperPAV information. You can specify a single device address, a range of device addresses, or any combination.

### Responses

Response 1:

If you enter QUERY PAV ALL, you see a response like this:

```
Device 0190 is a base Parallel Access Volume device with the following
aliases: 0192 0194 0196 0198
Device 0192 is an alias Parallel Access Volume device whose base device
is 0190
Device 0194 is an alias Parallel Access Volume device whose base device
is 0190
Device 0196 is an alias Parallel Access Volume device whose base device
is 0190
Device 0198 is an alias Parallel Access Volume device whose base device
is 0190
Device 01A0 is a base Parallel Access Volume device with the following
aliases: 01A4 01A8
Device 01A4 is an alias Parallel Access Volume device whose base device
is 01A0
Device 01A8 is an alias Parallel Access Volume device whose base device
is 01A0
Device 01D2 is a base Parallel Access Volume device with the following
aliases: 01D3 01E6 01EA
Device 01D3 is an alias Parallel Access Volume device whose base device
is 01D2
Device 01E6 is an alias Parallel Access Volume device whose base device
is 01D2
```

```

Device 01EA is an alias Parallel Access Volume device whose base device
is 01D2
Device 4300 is a base HyperParallel Access Volume device in Pool 2
Device 4341 is an alias HyperParallel Access Volume device in Pool 2
Device 4342 is an alias HyperParallel Access Volume device in Pool 2
Device 4343 is an alias HyperParallel Access Volume device in Pool 2
Device 4344 is an alias HyperParallel Access Volume device in Pool 2
Device 4345 is an alias HyperParallel Access Volume device in Pool 2
Device 8300 is a base HyperParallel Access Volume device in Pool 6
Device 8341 is an alias HyperParallel Access Volume device in Pool 6
Device 8342 is an alias HyperParallel Access Volume device in Pool 6
Device 8343 is an alias HyperParallel Access Volume device in Pool 6
Device 8344 is an alias HyperParallel Access Volume device in Pool 6
Device 8345 is an alias HyperParallel Access Volume device in Pool 6

```

#### Response 2:

If you enter QUERY PAV ALL when no PAV or HyperPav devices exist, you see a response like this:

```
There are no Parallel Access Volume devices known to this system
```

#### Response 3:

If you enter QUERY PAV *rdev* for a PAV alias device, you see a response like this:

```
Device rdev is an alias Parallel Access Volume device whose
base device is base_rdev
```

#### Response 4:

If you enter QUERY PAV *rdev* for a HyperPAV alias device, you see a response like this:

```
Device rdev is an alias HyperParallel Access Volume device in Pool nnnnn
```

#### Response 5:

If you enter QUERY PAV *rdev* for a PAV base device, you see a response like this:

```
Device rdev is a base Parallel Access Volume device with the
following aliases: alias1_rdev alias2_rdev
```

#### Response 6:

If you enter QUERY PAV *rdev* for a HyperPAV base device, you see a response like this:

```
Device rdev is a base HyperParallel Access Volume device in Pool nnnnn
```

#### Response 7:

If you enter QUERY PAV *rdev* for a device that is not a PAV or HyperPAV device, you see a response like this:

```
Device rdev is not a Parallel Access Volume device
```

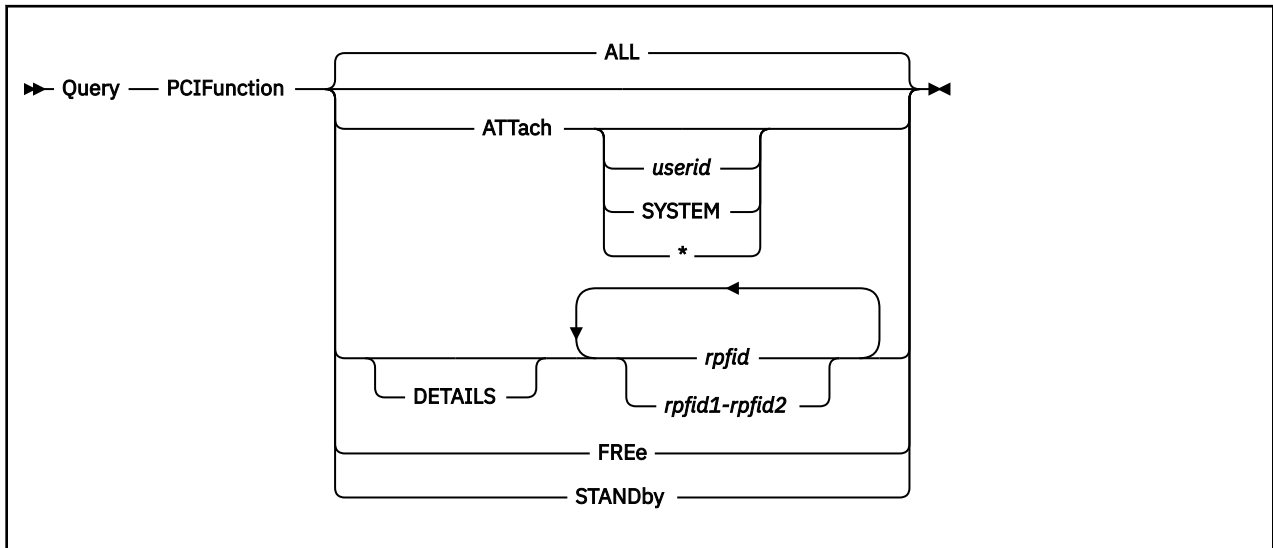
## Usage Notes

1. A VIRTUAL version of this command, QUERY VIRTUAL PAV, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL PAV” on page 1319](#).

## Messages

- HCP003E Invalid option - *{option | command contains extra option(s) starting with option}*
- HCP026E Operand missing or invalid
- HCP040E Device *rdev* does not exist

## QUERY PCIFUNCTION



### Authorization

Privilege Class: B

### Purpose

Use QUERY PCIFUNCTION to display the status of real PCI functions.

### Operands

#### ALL

indicates all real PCI functions. This is the default.

#### ATTach

indicates real PCI functions that are attached to a user or the system.

#### ATTach *userid*

indicates real PCI functions that are attached to a user. If you specify *userid*, only functions attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all functions attached to any user on the system.

#### ATTach SYSTEM

indicates real PCI functions that are attached to the system.

#### DETAILS *rpfid*

#### DETAILS *rpfid1-rpfid2*

displays detailed information about real PCI functions.

#### *rpfid*

#### *rpfid1-rpfid2*

is a real PCI function ID or a range of real PCI function IDs whose status is to be queried. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. The total number of real PCI functions specified cannot exceed 256.

#### FREe

indicates all real PCI functions that are available for use by the configuration and are not currently attached to a user or the system.

#### STANdby

indicates all real PCI functions currently in the standby state.

## Usage Notes

1. A VIRTUAL version of this command, QUERY VIRTUAL PCIFUNCTION, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL PCIFUNCTION”](#) on page 1322.

## Responses

Response 1:

If you enter QUERY PCIFUNCTION with the ATTACH operand, you will see something like:

```
PCIF 00000021 ATTACHED TO USER01 00000021 DISABLED 10GbE RoCE
PCIF 00000029 ATTACHED TO USER02 00000003 ENABLED 10GbE RoCE
```

The first number is the real PCI function ID. USER01 and USER02 are user IDs to which the PCI functions are attached. The next number is the virtual PCI function ID used by a virtual machine to refer to the PCI function. Following the virtual PCI function ID is the state of the PCI function and the type of PCI function.

The state of the function is one of the following:

### BLOCKED

PCI function blocked

### DISABLED

The indicated PCI function has not been initialized.

### ENABLED

The indicated PCI function has been detected and initialized by the guest.

### ERROR

Error detected by z/PCI facility

### PERMERR

Error recovery for PCI function unsuccessful

### STANDBY

PCI function not available to the configuration

Response 2:

If you enter QUERY PCIFUNCTION with the FREE operand, you will see something like:

```
PCIF 00000021 FREE DISABLED 10GbE RoCE
PCIF 00000029 FREE DISABLED 10GbE RoCE
```

The number is the real PCI function ID followed by FREE, the state of the PCI function and the type of PCI function.

Response 3:

If you enter QUERY PCIFUNCTION with the STANDBY operand, you will see something like:

```
PCIF 00000021 NOT CONFIGURED STANDBY 10GbE RoCE
PCIF 00000029 NOT CONFIGURED STANDBY 10GbE RoCE
```

The number is the real PCI function ID followed by NOT CONFIGURED, the state of the PCI function and the type of PCI function.

Response 4:

If you enter QUERY PCIFUNCTION with the ALL operand or a real PCI function ID or range of real PCI function IDs, you will see something like:

```
PCIF 00000003 ATTACHED TO USER01 00000001 DISABLED 10GbE RoCE
PCIF 00000004 FREE DISABLED 10GbE RoCE
PCIF 00000021 NOT CONFIGURED STANDBY 10GbE RoCE
PCIF 00000026 NOT CONFIGURED STANDBY 10GbE RoCE
PCIF 00000029 FREE DISABLED 10GbE RoCE
```

## QUERY PCIFUNCTION

```
PCIF 00000032 ATTACHED TO USER02 00000032 ENABLED 10GbE RoCE
PCIF 00000033 FREE ERROR 10GbE RoCE
```

Response 5:

If you enter QUERY PCIFUNCTION with the DETAILS operand, you will see something like:

```
PCIF 00000003, TYPE = 10GbE RoCE
PCHID = 3C0, VFN = 02, UID = 00001 PORT = --
NETID = NETIDPort1 NETIDPort2 -----
```

The first line of the response tells you the real PCI function ID and the type of PCI function. The lines that follow contain the details.

Details	Meaning
PCHID	physical channel identifier
VFN	virtual function number
UID	user-defined identifier value for a particular PCI function
PORT	port number
NETID	asserted physical-network identifiers corresponding to the order of the ports of the adapter. Dashes indicate no network ID has been assigned to the port.

Response 6:

If you enter QUERY PCIFUNCTION with the ALL operand and there are no PCI functions in the LPAR, you will receive the following response:

```
A PCI function was not found.
```

Response 7:

If you enter QUERY PCIFUNCTION for a PCI function that is associated with an NVMe EDEVICE, you will receive a response that is similar to the following example:

```
PCIF 00000071 ATTACHED TO SYSTEM EDEV 0F80 ENABLED NVMe
```

The first number (00000071) is the real PCI function ID, and it is attached to SYSTEM. The next number (0F80) is the address of the initial EDEVICE that is associated with the function. The PCI function state is ENABLED and its type is NVMe. A similar line would appear in the response to QUERY PCIFUNCTION with the ATTACH, ATTACH SYSTEM, or ALL operand. If the NVMe function is removed from the configuration, then the function state becomes STANDBY. If the function is subsequently added to the configuration, then the function state becomes DISABLED and the associated EDEVICES must be deleted and redefined to restore access to the EDEVICES.

If you enter QUERY PCIFUNCTION with the DETAILS option for an NVMe function that is associated with an EDEVICE, then additional details are provided unless any of the following conditions are true:

- The PCIe function was deconfigured or deleted.
- The PCIe function encountered a permanent error.
- The PCIe function reported a fatal status error.

The DETAILS response contains information about the device configuration, capacity, and health, and might look like the following example.

```
PCIF 00000071, TYPE = NVMe
PCHID = 0330, VFN = ---, UID = 00000, PORT = --
NETID = -----
Model Number: INTEL SSDPE2KX040T8
Serial Number: BTLJ906003YL4P0DGN
Firmware Version: VDV10131
PCI Vendor/Subsystem ID: 0x8086
```



```

IEEE OUI Identifier: 0x5CD2E4
Total NVM Capacity: 4000787030016
Unallocated NVM Capacity: 0
Controller ID: 0
Number of Namespaces: 1
Namespace 1 Size/Capacity: 4000787030016
Namespace 1 Formatted LBA Size: 512
Namespace 1 IEEE EUI-64: 0x5CD2E49BB23E0100
Firmware Updates: 1 Slot
Optional Admin Commands: 0x000E
Optional NVM Commands: 0x0006
Maximum Data Transfer Size: 32 Pages
Warning Comp. Temp. Threshold: 70 Celsius
Critical Comp. Temp. Threshold: 80 Celsius
Last device self-test result: Not Supported
Critical Warning: 0x00
Temperature: 30 Celsius
Available Spare: 100%
Available Spare Threshold: 10%
Percentage Used: 0%
Data Units Read: 35930574
Data Units Written: 24613697
Host Read Commands: 4445149911
Host Write Commands: 2709514387
Controller Busy Time: 289
Power Cycles: 76
Power On Hours: 14142
Unsafe Shutdowns: 73
Media and Data Integrity Errors: 0
Error Information Log Entries: 95
Warning Comp. Temperature Time: 0
Critical Comp. Temperature Time: 0

```

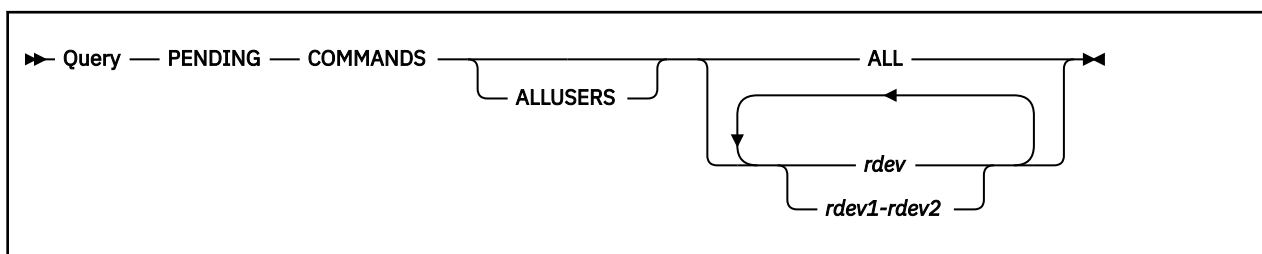
Error Information							
No	SQId	CmdId	Status	PELoc	LBA	NSId	VS
0	66	-	0xC00C	-	0	-	-
1	67	-	0xC00C	-	0	-	-
2	79	-	0xC00C	-	0	-	-
3	5	-	0xC00C	-	0	-	-
4	14	-	0xC00C	-	0	-	-
5	7	-	0xC00C	-	0	-	-
6	79	-	0xC00C	-	0	-	-
7	5	-	0xC00C	-	0	-	-
8	15	-	0xC00C	-	0	-	-
9	19	-	0xC00C	-	0	-	-
10	5	-	0xC00C	-	0	-	-
11	64	-	0xC00C	-	0	-	-
12	63	-	0xC00C	-	0	-	-
13	63	-	0xC00C	-	0	-	-
14	52	-	0xC00C	-	0	-	-
15	67	-	0xC00C	-	0	-	-
16	125	-	0xC00C	-	0	-	-
17	127	-	0xC00C	-	0	-	-
18	25	-	0xC00C	-	0	-	-
19	42	-	0xC00C	-	0	-	-

For more information about NVMe specifications, see [NVM Express \(nvmexpress.org\)](http://nvmexpress.org).

## Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP021E A real PCI function ID was not supplied or it is invalid.
- HCP040E PCI function *rpfid* does not exist
- HCP1003E *rpfid* not found.
- HCP5780E Adjunct does not support the *command subcommand* command
- HCP6000E The range of PCI function ids cannot exceed 256.

## QUERY PENDING COMMANDS



### Authorization

Privilege Class: B

### Purpose

Use QUERY PENDING COMMANDS to display the commands that you have entered and, optionally, that others have entered for which the associated asynchronous function has not yet completed.

### Operands

#### ALLUSERS

is an optional keyword that indicates that all pending commands for the specified devices are to be displayed, including pending commands that were not issued by you.

#### ALL

indicates that pending commands for all devices are to be displayed.

#### *rdev*

#### *rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers of devices in the subsystems being queried.

### Usage Notes

1. Many CP commands associated with DASD subsystems result in a channel program being issued to a device or subsystem. In most cases, these operations complete in a very short time, and the command response is presented to the operator immediately. However, for some cached DASD subsystems, some of these operations do not complete immediately. Use the QUERY PENDING COMMANDS command to display those commands that you have issued that have not yet completed. See [“Delayed Responses to CP Commands”](#) on page 7 for further information and an example.

### Responses

Response 1:

```
No pending commands exist.
```

This response is displayed if QUERY PENDING COMMANDS ALL is issued and there are no pending commands.

Response 2:

```
No pending commands exist for rdev.
```

This response is displayed if there are no pending commands for the indicated device.

**rdev**

is the real device number of a device being queried.

Response 3:

```
Command pending:  command rdev.
```

This response is repeated for each pending command for each specified device.

**command**

is the command for which the associated asynchronous function is not yet complete.

**rdev**

is the real device number of a device being queried.

Response 4:

```
Command pending for userid:  command rdev.
```

This response is repeated for each pending command for each specified device for commands not issued by you.

**userid**

is the ID of the user who issued the command.

**command**

is the command for which the associated asynchronous function is not yet complete.

**rdev**

is the real device number of a device being queried.

**Messages**

- HCP009E Invalid range - *range*
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist

## QUERY PER

---

### **Authorization**

Privilege Class: G

### **Purpose**

QUERY PER is recognized as a synonym for QUERY TRACE.

Within VM, TRACE monitors events that occur in your virtual machine. Please refer to the CP QUERY TRACE command.

## QUERY PFnn



### Authorization

Privilege Class: G

### Purpose

Use QUERY PFnn to display the program function key assignments.

### Operands

**PF**

**PFnn**

requests a display of the program function command lines associated with the program function key you designate by the value *nn*. The value *nn* is a number from 1 (or 01) through 24. If you do not specify a value for *nn*, all program function keys and their associated data lines are displayed.

### Usage Notes

1. See [“SET PFnn” on page 1602](#) for an explanation of the way to define and use program function keys.

### Responses

Response 1:

```
PFnn [SUBSTITU]{DELAY } pfdata
           {IMMED }
           {NODISP}
```

**PFnn**

is the program function key number.

**SUBSTITU**

indicates that tokens from the input are substituted in *pfdata* when the PF key is pressed. Otherwise, no substitution takes place.

**DELAY**

**IMMED**

**NODISP**

indicates when the program function is to be run. IMMED means the program function is run when you press the PF key. DELAY means the program function is displayed in the input area when you press the PF key. NODISP means the program function is executed when you press the PF key, but the function is not redisplayed on the terminal.

***pfdata***

is the program function command line associated with the key.

Response 2:

If the program function key has been set to retrieve prior commands, you can receive these responses:

```
PFnn RETRIEVE BACKWARD
PFnn RETRIEVE FORWARD
```

## QUERY PFnn

```
PFnn RETRIEVE BACKWARD n  
PFnn RETRIEVE FORWARD n
```

Response 3:

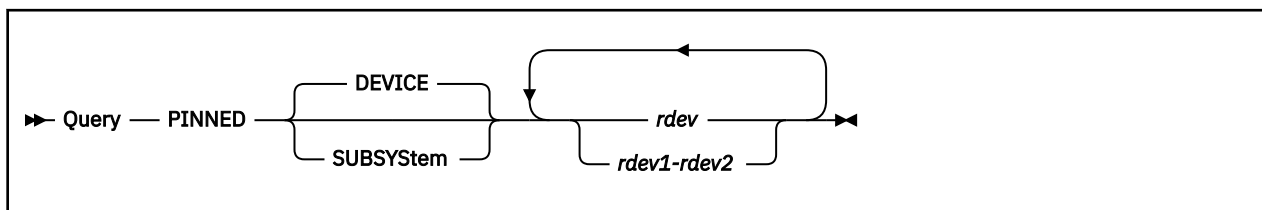
If there is no function defined for the program function key, you see the following:

```
PFnn UNDEFINED
```

### Messages

- HCP003E Invalid option - *option*
- HCP263E Too many operands were supplied for this command.

## QUERY PINNED



### Authorization

Privilege Class: B

### Purpose

Use QUERY PINNED to display the data tracks of one or more DASD that have become pinned in subsystem storage (cache) or nonvolatile storage.

This command may also be used to determine the total amount of subsystem storage (cache) or nonvolatile storage that contains pinned data.

### Operands

#### DEVICE

indicates that pinned tracks for the specified devices are to be displayed. This is the default value.

#### SUBSYSTEM

indicates that the total amount of subsystem cache and nonvolatile storage containing pinned data is to be displayed.

#### *rdev*

#### *rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers of devices in the subsystems being queried.

### Usage Notes

1. This command may be issued to offline devices.

### Responses

Response 1:

The following response is displayed for each specified device that does not have any pinned tracks when you enter QUERY PINNED DEVICE:

```
rdev No pinned tracks exist for this device.
```

#### *rdev*

is the real device number of a device being queried.

Response 2:

The following response is displayed for each specified device when you enter QUERY PINNED DEVICE:

```
rdev Pinned track data:
tt-cccchh tt-cccchh tt-cccchh tt-cccchh
tt-cccchh tt-cccchh tt-cccchh tt-cccchh
```

## QUERY PINNED

### **rdev**

is the real device number of a device being queried.

### **tt**

is the type of pinned track.

### **01**

indicates that the operation can be retried when the hardware is repaired.

### **02**

indicates that the operation cannot be retried because of a format error.

### **04**

indicates that the data is pinned in NVS, the operation can be retried.

### **08**

indicates that the cache copy of the data is defective.

### **cccchh**

is the address (cylinder, head) of a data track pinned in subsystem cache or nonvolatile storage.

Response 3:

This response is repeated for each *unique subsystem* identified by the devices specified on the command line when you enter QUERY PINNED SUBSYSTEM:

```
rdev Pinned subsystem data:  
cccccccK Bytes are in cache.  
nnnnnnnK Bytes are in non-volatile storage.
```

### **rdev**

is the real device number of one of the specified devices in a subsystem being queried.

### **ccccccc**

is the total amount of cache storage containing pinned data.

### **nnnnnnnn**

is the total amount of nonvolatile storage containing pinned data.

Response 4:

This response is repeated for each *unique subsystem* identified by the devices specified on the command line when you enter QUERY PINNED SUBSYSTEM if the amount of pinned data in nonvolatile storage cannot be determined.

```
rdev Pinned subsystem data:  
cccccccK Bytes are in cache.  
          Bytes in non-volatile storage cannot be determined.
```

### **rdev**

is the real device number of one of the specified devices in a subsystem being queried.

### **ccccccc**

is the total amount of cache storage containing pinned data.

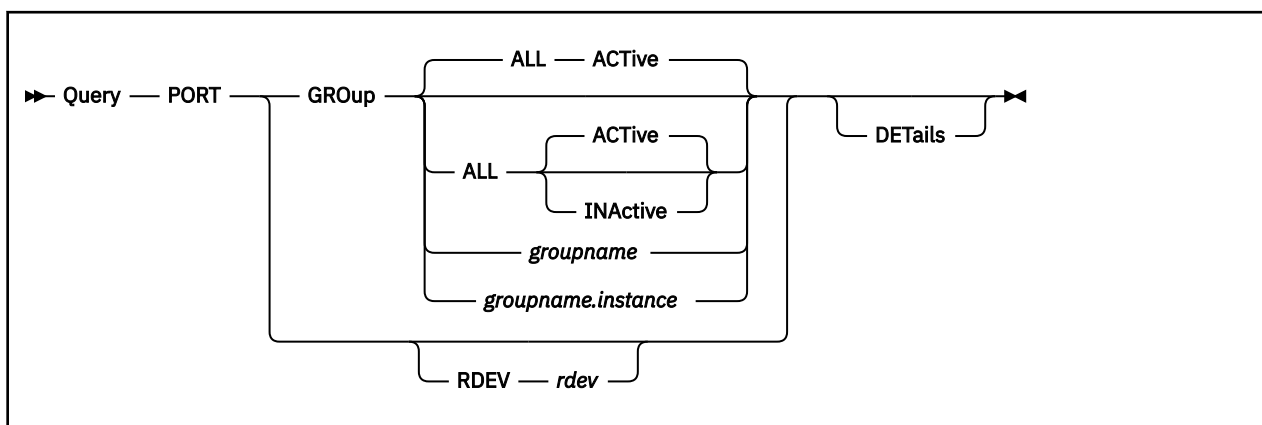
## Messages

- HCP006E Invalid device type - {rdev/vdev/ldev}
- HCP009E Invalid range - range
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP319E Error processing subsystem command - cmd for DASD rdev
- HCP332E Invalid control unit type - rdev



- HCP1109E The *command* command cannot be processed for device *rdev* because the device is the secondary in a duplex pair.
- HCP1120E The requested [command|DETACH|GIVE] for device [*ldev*|*rdev*] did not complete in the allotted time.

## QUERY PORT



### Authorization

Privilege Class: B

### Purpose

Use the QUERY PORT command to display information about link aggregation groups or devices that have been defined for virtual switches on the system.

### Operands

#### GROUP ALL

indicates a request for all active port groups defined in the system. This is the default when an explicit GROUP *groupname* is omitted.

#### ACTive

indicates a request to return only those groups that have been associated with an active virtual switch.

#### INActive

indicates a request to return only those groups that have not been associated with a virtual switch.

#### GROUP *groupname*

indicates a request for the specified *groupname*.

#### GROUP *groupname.instance*

indicates a request for the specific instance (0-3) of a *groupname*. This format can be used to display a specific instance of a shared port group when it is configured to a global virtual switch on a system.

#### RDEV *rdev*

indicates a request for the specified real device.

#### DETAils

indicates a request for full information about the groups or device specified. This will include Link Aggregation Control Protocol (LACP) information for the selected group or device. For shared port groups, the display includes information about how each copy of the shared port group is configured on each IVL domain member.

### Examples

1. QUERY PORT GROUP *groupname*:

Display a list of information about a group and the associated devices that have been defined for the group where the group has not been associated with a virtual switch:

```
cp q port gro prodgrp

Group: PRODGRP   Inactive   LACP Mode: Active   Exclusive
VSWITCH <none>                               ifIndex: 1
Load Balancing: Independent   Interval: 300
RDEV: 3300.P00   Adapter ID: 29640007E386.0250
RDEV: 3400.P00   Adapter ID: 29640007E386.0251
RDEV: 3500.P00   Adapter ID: 29640007E386.0252
RDEV: 3600.P00   Adapter ID: 29640007E386.0254
```

## 2. QUERY PORT GROUP *groupname*:

Display a list of information about a group and the associated devices that have been defined for the group where the group has been associated with an active virtual switch:

```
cp query port gro ethgrp

Group: ETHGRP   Active   LACP Mode: Active   Exclusive
VSWITCH SYSTEM ETH1                               ifIndex: 1
Load Balancing: Independent   Interval: 300
RDEV: 9C00.P00 VDEV: 9C00 Controller: DTCVSW1   ACTIVE
Adapter ID: 29640007E386.0260
Uplink Port Connection:
Data Counters Cleared: 11/12/15   13:54:00
Discarded Counters Cleared: 11/12/15   13:54:00
Error Counters Cleared: 11/12/15   13:54:00
MAC address: 02-22-90-00-00-05
RX Packets: 190   Discarded: 0   Errors: 0
TX Packets: 18   Discarded: 0   Errors: 0
RX Bytes: 2101   TX Bytes: 2573
Device: 9C02 Unit: 002 Role: DATA Port: 0001
Partner Switch Capabilities: No_Reflective_Relay
LAG Port Controller: Active
RDEV: 9D00.P01 VDEV: 9D00 Controller: DTCVSW1   ACTIVE
Adapter ID: 29640007E386.0560
Uplink Port Connection:
Data Counters Cleared: 11/12/15   13:54:00
Discarded Counters Cleared: 11/12/15   13:54:00
Error Counters Cleared: 11/12/15   13:54:00
MAC address: 02-22-90-00-00-06
RX Packets: 19   Discarded: 0   Errors: 0
TX Packets: 8   Discarded: 0   Errors: 0
RX Bytes: 152   TX Bytes: 1280
Device: 9D02 Unit: 002 Role: DATA Port: 0002
Partner Switch Capabilities: No_Reflective_Relay
LAG Port Controller: Active
```

## 3. QUERY PORT RDEV *rdev*:

Display information about a device that has been associated with an active virtual switch:

```
cp query port rdev 9d00

Group: ETHGRP   Active   LACP Mode: Active   Exclusive
VSWITCH SYSTEM ETH1                               ifIndex: 1
Load Balancing: Independent   Interval: 300
RDEV: 9D00.P01 VDEV: 9D00 Controller: DTCVSW1   ACTIVE
Adapter ID: 29640007E386.0560
Uplink Port Connection:
MAC address: 02-22-90-00-00-06
RX Packets: 19   Discarded: 0   Errors: 0
TX Packets: 9   Discarded: 0   Errors: 0
RX Bytes: 152   TX Bytes: 1440
Device: 9D02 Unit: 002 Role: DATA Port: 0002
Partner Switch Capabilities: No_Reflective_Relay
LAG Port Controller: Active
```

## 4. QUERY PORT GROUP *groupname* DETAILS:

## QUERY PORT

Display a list of detailed information including LACP activity about a group and the associated devices that have been defined for the group.

```
q port group ethgrp details

Group: ETHGRP Active LACP Mode: Active Exclusive
VSWITCH SYSTEM ETH1 ifIndex: 1
Load Balancing: Independent Interval: 300
GROUP Information:
  PORT Information - Total Frames per Interval:
  Device Status Previous Current
  -----
  9C00 Active 90 82
  9D00 Active 5 2
  ROUTING Information - Frame Distribution per Interval:
  MAC Device Previous Current
  0 9C00 4 0
  1 9D00 0 0
  2 9C00 0 0
  3 9D00 0 0
  4 9C00 2 0
  5 9D00 2 0
  6 9C00 0 0
  7 9D00 0 0
RDEV: 9C00.P00 VDEV: 9C00 Controller: DTCVSW1 ACTIVE
Adapter ID: 29640007E386.0260
Uplink Port Connection:
MAC address: 02-22-90-00-00-05
RX Packets: 248 Discarded: 0 Errors: 0
TX Packets: 90 Discarded: 0 Errors: 0
RX Bytes: 20192 TX Bytes: 6486
Device: 9C02 Unit: 002 Role: DATA Port: 0001
Partner Switch Capabilities: No_Reflective_Relay
LAG Port Controller: Active
PROTOCOL Counters:
LACP RX: 43 Marker RX: 0
LACP TX: 10 Marker TX: 5 Timeouts: 0
ACTOR Information:
System ID: 32768,02-22-90-00-00-02 Oper Key: 2
Port Priority: 32768 Port: 0001 Group Key: 2
State: 3D - LACP_Active Slow AGG SYNC DIST COLL
PARTNER Information:
System ID: 23456,00-13-80-FD-91-00 Oper Key: 1
Port Priority: 32768 Port: 0005 Group Key: 1
State: 3D - LACP_Active Slow AGG SYNC DIST COLL
RDEV: 9D00.P01 VDEV: 9D00 Controller: DTCVSW2 ACTIVE
Adapter ID: 29640007E386.0560
Uplink Port Connection:
MAC address: 02-22-90-00-00-06
RX Packets: 22 Discarded: 0 Errors: 0
TX Packets: 13 Discarded: 0 Errors: 0
RX Bytes: 2986 TX Bytes: 1806
Device: 9D02 Unit: 002 Role: DATA Port: 0002
Partner Switch Capabilities: No_Reflective_Relay
LAG Port Controller: Active
Unicast IP Addresses:
9.60.8.1 MAC: 00-09-12-F3-40-20 Remote
PROTOCOL Counters:
LACP RX: 9 Marker RX: 0
LACP TX: 9 Marker TX: 0 Timeouts: 0
ACTOR Information:
System ID: 32768,02-22-90-00-00-02 Oper Key: 2
Port Priority: 32768 Port: 0002 Group Key: 2
State: 3D - LACP_Active Slow AGG SYNC DIST COLL
PARTNER Information:
System ID: 23456,00-13-80-FD-91-00 Oper Key: 1
Port Priority: 32768 Port: 0004 Group Key: 1
State: 3D - LACP_Active Slow AGG SYNC DIST COLL
Backup Devices:
RDEV: 2100.P00 VDEV: 2100 Controller: DTCVSW2 BACKUP
RDEV: 1E0A.P00 VDEV: 1E0A Controller: DTCVSW1 BACKUP
RDEV: 1E0E.P00 VDEV: 1E0E Controller: DTCVSW2 BACKUP
```

### 5. QUERY PORT GROUP BLUE DETAILS:

Display a list of information about a shared port group, including members configured on other systems.

```

cp query port group BLUE details

Group: BLUE.0 Active LACP Mode: Active Shared
VSWITCH CASEY.RICK ifIndex: 1
Load Balancing: Collaborative Interval: 300 CASEY.RICK
GROUP Information:
PORT Information - Total Frames per Interval:
      Device Status Local Previous Local Current Total Previous
      -----
      0400 Active 90 82 206
      0500 Active 5 2 67890
      -----
      Total Port Group Frames: 68096
      Last Load Balance: GDLSADP1 Date: 03/30/16 Time: 10:02:55

ROUTING Information - Frame Distribution per Interval:
      MAC Device Previous Current
      ---
      0 0400 4 0
      1 0500 0 0
      2 0400 0 0
      3 0500 0 0
      4 0400 2 0
      5 0500 2 0
      6 0400 0 0
      7 0500 0 0
RDEV: 0400.P00 VDEV: 0400 Controller: DTCVSW1
Adapter ID: 29640007E386.0560
Uplink Port Connection:
MAC address: 02-22-90-00-00-05
RX Packets: 248 Discarded: 0 Errors: 0
TX Packets: 90 Discarded: 0 Errors: 0
RX Bytes: 20192 TX Bytes: 6486
Device: 0400 Unit: 000 Role: DATA Port: 0001
Partner Switch Capabilities: No_Reflective_Relay
LAG Port Controller: Active
PROTOCOL Counters:
LACP RX: 43 Marker RX: 0
LACP TX: 10 Marker TX: 5 Timeouts: 0
ACTOR Information:
System ID: 32768,02-22-90-00-00-02 Oper Key: 2
Port Priority: 32768 Port: 0001 Group Key: 2
State: 3D - LACP_Active Slow AGG SYNC DIST COLL
PARTNER Information:
System ID: 23456,00-13-80-FD-91-00 Oper Key: 1
Port Priority: 32768 Port: 0005 Group Key: 1
State: 3D - LACP_Active Slow AGG SYNC DIST COLL
RDEV: 0500.P01 VDEV: 0500 Controller: DTCVSW1
Adapter ID: 29640007E386.01C8
Member: CASEY
Scope: Synchronized
LAG Synchronization token: CDE60F8EA6AD18AC
Mode: Connected
Member: JONES
Scope: Synchronized
LAG Synchronization token: CDE60F8EA6AD18AC
Mode: Inactive

```

## 6. QUERY PORT GROUP ALL ACTIVE

Display a list of information for all active port groups. For shared port group, all instances will be displayed.

```

cp query port group all active

Group: BLUE.0 Active LACP Mode: Active Shared
VSWITCH CASEY.RICK ifIndex: 2112
Load Balancing: Collaborative Interval: 300
RDEV: 7240.P00 VDEV: 7240 Controller: DTCVSW1 ACTIVE
Adapter ID: 29640007E386.0260
Group: BLUE.1 Active LACP Mode: Active Shared
VSWITCH CASEY.SUE ifIndex: 2112
Load Balancing: Collaborative Interval: 300
RDEV: 7243.P00 VDEV: 7243 Controller: DTCVSW1 ACTIVE
Adapter ID: 29640007E386.0260

```

## QUERY PORT

```
Group: BLUE.2 Active LACP Mode: Active Shared  
VSWITCH CASEY.SAM ifIndex: 2112  
Load Balancing: Collaborative Interval: 300  
RDEV: 7246.P00 VDEV: 7246 Controller: DTCVSW1 ACTIVE  
Adapter ID: 29640007E386.0260
```

### Responses

Response 1: The QUERY PORT response includes optional lines. The general format is:

#### **Group *groupname* [*instance*]**

*groupname* identifies the link aggregation group. When a shared port group is in use on a global virtual switch, the group name is displayed with an instance number in the format *groupname.instance*. This is most useful when the shared port group is in use by more than one global virtual switch on a single system. For example, when RICK and SAM are global virtual switches defined on the same system and both are configured to use shared port group Blue, BLUE.0 is displayed for the line describing RICK and BLUE.1 is displayed for the line describing SAM.

#### **LACP Mode**

identifies whether the Link Aggregation Control Protocol (LACP) mode is Active or Inactive.

#### **VSWITCH <none> | SYSTEM *switchname systemname.switchname***

identifies the virtual switch that has been associated with the group. If <none> is listed, an association has not been made. For GLOBAL virtual switches, *systemname.switchname* is shown in place of SYSTEM *switchname* where *systemname* identifies the 8 byte EBCDIC z/VM System Identifier of the system where a copy of the shared port group exists and is in use on a global virtual switch.

#### **Interval**

*nnnn* indicates the length in time in seconds between load balancing intervals.

#### **ifIndex**

*nnnn* is the ifIndex representing the aggregator for the port group.

#### **Load Balancing**

displays information about load balancing.

##### **Independent**

Each VSwitch sharing the same LAG will only balance the workload on its VSwitch. There is no consideration for the other VSwitches also sharing the same LAG.

##### **Forced Independent**

The port group is configured to use the Collaborative Load Balance management method, but is being forced by z/VM to use the Independent Load Balance management method. All z/VM Systems currently connected to the same IVL are not capable or are not able to functionally support Collaborative Load Balance.

##### **Collaborative**

All VSwitches sharing the same LAG will collaborate with each other to balance the LAG's workload.

##### **Interval**

*nnnn* indicates the length in time in seconds between load balancing intervals.

#### **RDEV**

*nnnn.Pnn* indicates the base real device number of a QDIO device defined for the virtual switch. The base number is the first of three consecutive device numbers and OSA-Express port number used to establish a QDIO connection.

#### **VDEV**

*nnnn* indicates the base virtual device number used by the virtual switch controller for the RDEV.

#### **Controller**

is the user ID of the TCP/IP virtual machine acting as a controller for the RDEV.

#### **Adapter ID**

describes the OSA-Express port in use on system *systemname*. The format is:

```
cccccccccccc.pchid
```

**cccccccccccc**

represents the 12 bytes of text character and hexadecimal information that uniquely identifies the processor.

**pchid**

identifies the Physical Channel ID representing the OSA-Express feature.

**Status**

indicates the status of the RDEV. If the field is blank, the device is operating in the ACTIVE state and operating normally. Possible states are:

**ACTIVE (displayed as blank)**

The port is currently available and in use within the link aggregation group.

**Waiting**

The port is available but is not currently being used within the link aggregation group.

**Suspended**

All data transmissions are suspended until pending requests are complete or they can be moved to another port within the link aggregation group.

**Error**

The port is not available to the link aggregation group. The reason for the error is listed in the response.

**Reason**

Possible error reasons are:

**Pending QDIO Activation**

The device has been assigned to a controller and is completing initialization.

**Pending Routing Assignment**

The device is being activated and will become available during the next cycle.

**Port inoperable**

The device is suspended.

**ABEND occurred**

An abend occurred while processing data for this device.

**Pending fallback**

The device is failing back to the link aggregation group.

**LACP negotiation**

The device is waiting for LACP port negotiation to add this port to the group. Inbound datagrams are accepted.

**Partner LACP information mismatch**

The partner system ID and/or key do not match the virtual switch information.

**Inoperable by LACP**

The partner has indicated the device is not part of the group.

**LACP not enabled on partner**

LACP is not currently enabled on the partner switch.

Response 2: In addition to the lines displayed in response 1, the QUERY PORT GROUP response will include the following lines when DETAILS are requested for a shared port group.

**Member: *systemname***

identifies the 8 byte EBCDIC z/VM system identifier of the domain member.

**Scope****Synchronized**

The local system's view of the shared port group matches the port group configuration on other IVL domain members.

**Pending Synchronized**

The local system is checking to make sure that configuration of the shared port group matches the existing configuration on all systems in the IVL domain.

### **Not Synchronized**

The local system's view of the shared port group does not match the configuration on other IVL domain members.

### **Error [msgno]**

The local system has identified a configuration mismatch with the way the shared port group is configured on another system in the IVL domain.

A message number may also be displayed to identify a specific synchronization problem.

### **LAG Synchronization Token**

A token representing the most recent update made to the shared port group. This token is for use by IBM personnel for diagnosing synchronization problems.

### **Mode: [Connected | Inactive]**

Connected indicates the port group is associated with a global VSwitch and the global VSwitch is connected. Inactive indicates the port group is not associated with a global VSwitch or the associated global VSwitch is not connected.

### **Take-over Status Member: *systemname* MAC address: *mac***

An additional line(s) shows take over status when the shared port group is involved in MAC address takeover to maintain connectivity following a failure.

A different global VSwitch member had an UPLINK port failure, and its MAC addresses are now assigned to one of the local OSAs on this global VSwitch member. The member name may be omitted if it cannot be obtained (for example if the member left the IVL domain). This is followed by a list of MAC Addresses.

## **Messages**

- HCP2837E No {active|inactive} groups found.
- HCP2783E GROUP *groupname* does not exist.
- HCP2783E RDEV *vdev* does not exist.
- HCP3185I The displayed response may be incomplete. The system is not joined with the IVL Domain.



## QUERY (Printer ID)

---

► Query — *ptid* ◄

### Authorization

Privilege Class: B

### Purpose

Use QUERY (Printer ID) to display the status of a virtual machine that controls a print server printer. The operand *ptid* is 1- to 8- alphanumeric characters. If you specify *ptid* as 4 or fewer characters, it must not contain only hexadecimal characters; because CP interprets *ptid* as *rdev*.

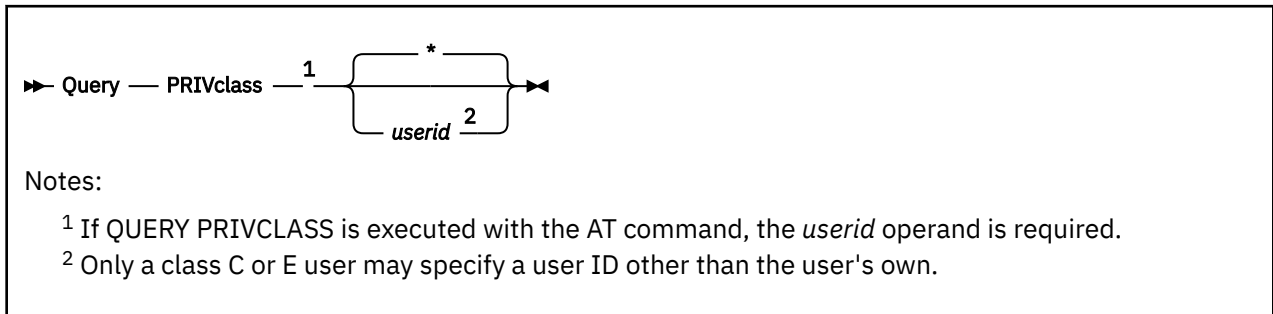
### Responses

The virtual machine controlling the printer issues responses to this command, not CP.

### Messages

None.

## QUERY PRIVCLASS



### Authorization

Privilege Class: Any, C, E

### Purpose

Use QUERY PRIVCLASS to display the current and directory-specified privilege classes available to a virtual machine. If you are a class C or E user, you can display the privilege classes for any virtual machine. If you are not a class C or E user, you can display the privilege classes only for your own virtual machine.

### Operands

\*

displays the privilege classes of the user who issues the command. This is the default.

*userid*

identifies the user whose privilege classes are to be displayed.

### Usage Notes

1. When you log on, CP sets your user ID's privilege classes to the privilege classes granted to you in your user directory entry.
2. If a privileged DirMaint user updates your directory entry by adding additional privilege classes while you are logged on, you can put those new privilege classes into effect by issuing the SET PRIVCLASS \* RESET command.

For more information, see [“SET PRIVCLASS” on page 1616](#).

### Responses

Response 1:

If you enter a QUERY PRIVCLASS command, you will see something like this:

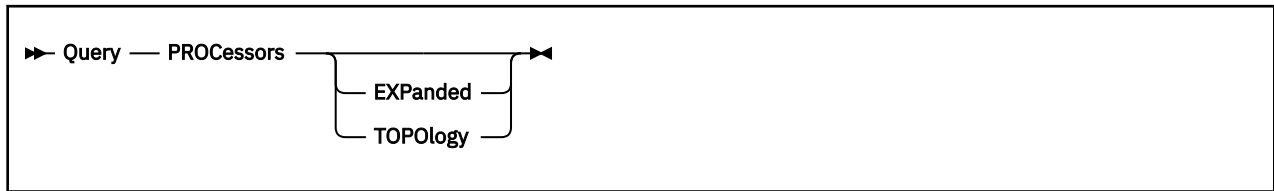
```
Privilege classes for user MINNIE
  Currently: BCEG
  Directory: ABCDEFG
The privilege classes are not locked against changes.
```

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP007E Invalid *userid* - *userid*
- HCP045E *userid* not logged on

- HCP052E Error in CP directory
- HCP053E *userid* not in CP directory

## QUERY PROCESSORS



### Authorization

Privilege Class: A, B, C, E

### Purpose

Use QUERY PROCESSORS to display a list of all real processors, and indicate the way in which each processor is being used. It also can be used to display the mode of the logical partition.

Use QUERY PROCESSORS TOPOLOGY to display the CPU topology. The topology is a tree representation that contains information about the specific logical CPUs that have affinity within the corresponding part of the real machine.

When multithreading is enabled, these commands also show the relationship between CPUs (threads) and the cores on which they reside.

### Operands

#### EXPanded

requests an expanded response.

#### TOPOlogy

requests current topology.

### Usage Notes

1. A primary processor is a processor with a CPU type that matches the type of the IPL processor. In an all-IFL configuration, each IFL processor is a primary processor. In all other configurations, primary processors are type CP. A secondary processor is a processor with a type that does not match the type of the IPL processor.
2. A PARKED processor can exist in vertical-polarization mode only.
3. When VM is running second-level, the CPU topology is not available.
4. When multithreading is enabled, gaps might occur in the sequence of processor addresses for cores that have less than the maximum number of activated threads. For an example, see [“QUERY MULTITHREAD”](#) on page 1026.
5. Unlike QUERY PROCESSORS and QUERY PROCESSORS EXPANDED, QUERY PROCESSORS TOPOLOGY does not include information on standby processors because they do not have an assigned placement within the topology.
6. Processors defined to the partition but outside the addressable range (0000 - 004F) might appear in monitor output, but will not appear in the QUERY PROCESSORS display.

### Responses

Response 1:

This is the response format when *no* operands are specified. Each processor's information is displayed as one of the following lines.

```
PROCESSOR nnnn MASTER type [CORE cccc]
PROCESSOR nnnn ALTERNATE type [CORE cccc]

PROCESSOR nnnn PARKED type [CORE cccc]
PROCESSOR nnnn STANDBY type [CORE cccc]
```

***nnnn***

is the logical processor number.

**MASTER**

indicates the master processor.

**ALTERNATE**

indicates an alternate processor (a nonparked processor other than the master).

**PARKED**

indicates a processor that is online, but CP has chosen not to run any work on it. A processor can be parked in vertical-polarization mode only.

**STANDBY**

indicates a processor that is part of the configuration but is not online. A STANDBY processor can be brought online by using the VARY PROCESSOR command.

***type***

indicates the type of the real processor. Type can be as follows:

**CP**

indicates the processor is a central processor (that is, a general-purpose processor).

**IFL**

indicates the processor is an IBM Integrated Facility for Linux processor.

**ZIIP**

indicates the processor is an IBM z Integrated Information Processor.

**ICF**

indicates the processor is an Internal Coupling Facility processor.

**CORE *cccc***

when multithreading is enabled, identifies the core ID of the logical core that contains the specified logical processor.

***userid***

is the user who owns the virtual CPU.

***mm***

is the virtual address of the virtual CPU.

Response 2:

This is the response format when the EXPANDED operand is specified. Each processor's information is displayed as one of the following first four lines.

```
PROCESSOR nnnn MASTER type [CORE cccc]
PROCESSOR nnnn ALTERNATE type [CORE cccc]

PROCESSOR nnnn PARKED type [CORE cccc]
PROCESSOR nnnn STANDBY type [CORE cccc]
PARTITION MODE {ESA/390 }
                {GENERAL }
                {LINUX-ONLY}
                {Z/VM }
```

where PARTITION MODE is the mode of the logical partition as specified in the image profile and the other fields are as described in Response 1.

Response 3:

## QUERY PROCESSORS

This is the response format when the TOPOLOGY operand is specified. Depending on the machine, one or two nesting levels might be displayed. The PROCESSOR line is repeated for each processor within the nesting level.

```
TOPOLOGY
NESTING LEVEL: nl ID: id
NESTING LEVEL: nl ID: id
PROCESSOR nnnn proctype type pe dv [CORE cccc]
PROCESSOR nnnn proctype type pe dv [CORE cccc]
```

### ***nl***

is the nesting level.

### ***id***

is the identifier for the nesting level.

### ***nnnn***

is the logical processor number.

### ***proctype***

can be MASTER, ALTERNATE, or PARKED, as described in Response 1.

### ***type***

indicates the type of the real processor, as described in Response 1.

### ***pe***

indicates the polarization and entitlement of a CPU.

### **HZ**

indicates that the CPU is polarized horizontally. Horizontally-polarized CPUs (of a given CPU type) receive approximately equal access to CPU resources.

### **VL**

indicates that the CPU is polarized vertically with low entitlement. Vertical-low CPUs are not entitled to any CPU resource and consume spare cycles only, if they are available.

### **VM**

indicates that the CPU is polarized vertically with medium entitlement. Vertical-medium CPUs are entitled to a fraction of a real CPU, but can consume more than this if there are spare cycles on the machine.

### **VH**

indicates that the CPU is polarized vertically with high entitlement. Vertical-high CPUs are entitled to an entire real CPU.

### ***dv***

identifies the four-digit decimal dispatch vector (work queue) membership of the processor. When two or more processors have the same *dv* value, they are selecting work from the same queue.

### **CORE *cccc***

when multithreading is enabled, identifies the core ID of the core that contains the specified processor.

Response 4:

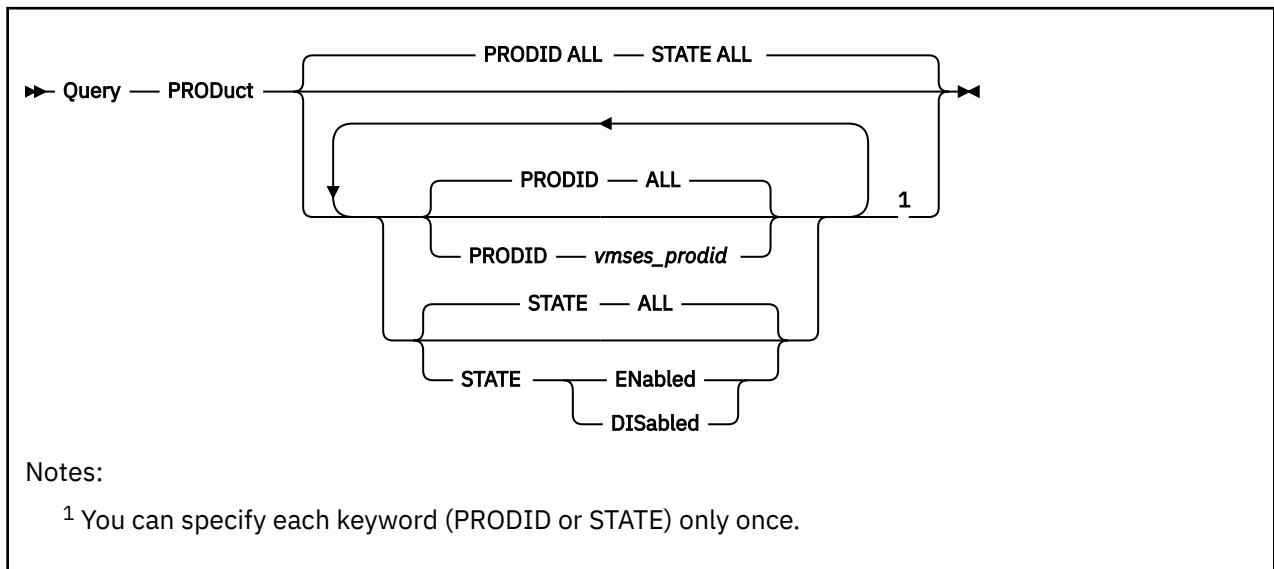
This is the response format when the TOPOLOGY operand is specified and topology information is not available.

```
CPU topology is not available.
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP013E Conflicting option - *option*
- HCP263E Too many operands were supplied for this command.

## QUERY PRODUCT



### Authorization

Privilege Class: C, E

### Purpose

Use QUERY PRODUCT to display information regarding the PRODUCT statements defined to the system in the system configuration file at IPL and defined using the SET PRODUCT command.

### Operands

#### PRODIG ALL

tells CP to display information about all product IDs defined to the system. This is the default.

#### PRODIG *vmses\_prodid*

is the identifier used by VMSES/E to install and service the given product. *vmses\_prodid* must be a 7 or 8 character identifier.

#### STATE ALL

tells the system to display the associated product information regardless of the current state. This is the default.

#### STATE ENABLED

displays information about the product IDs that are enabled on the system.

#### STATE DISABLED

displays information about product IDs that are explicitly disabled on the system.

### Usage Notes

1. QUERY PRODUCT commands gathers information about the current PRODUCT statements defined to the system. Although this information may be contained in the system configuration file, it may have been changed with a SET PRODUCT command since that time.
2. The output from QUERY PRODUCT is sorted by product ID from lowest to highest.
3. QUERY PRODUCT is not intended to be a programming interface. Programs requiring enablement data should use DIAGNOSE code X'27C'.

## QUERY PRODUCT

### Responses

Response 1:

To query a single product ID (5735FALQ), enter the following command:

```
QUERY PRODUCT 5735FALQ
```

```
Product State Description
5735FALQ Enabled TCP/IP LEVEL 310 - TCP/IP FEATURE (BASE)
```

Response 2:

To query ONLY enabled product IDs on the system, enter the following command:

```
QUERY PRODUCT STATE ENABLED
```

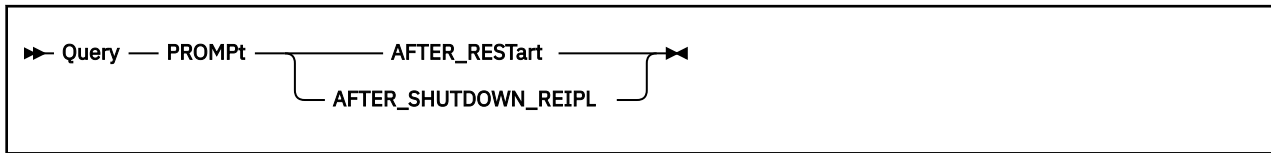
```
Product State Description
5735FALQ Enabled TCP/IP LEVEL 310 - TCP/IP FEATURE (BASE)
5877-00 Enabled Product ABC
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*.
- HCP0013E Conflicting option - *operand*
- HCP2768E Missing {product|product state}
- HCP6704E Missing token at end of line
- HCP6706E Invalid product {*string*|state *string*}
- HCP6720I No items found to satisfy selection criteria



## QUERY PROMPT



### Authorization

Privilege Class: B

### Purpose

Use QUERY PROMPT to display whether CP will prompt the operator for startup information following a CP restart after an abend or a shutdown REIPL.

### Operands

#### AFTER\_REStart

specifies to display whether CP will prompt the operator for startup information after a CP restart after an abend or not.

#### AFTER\_SHUTDOWN\_REIPL

specifies to display whether CP will prompt the operator for startup information after a shutdown REIPL or not.

### Usage Notes

1. Use the SET PROMPT command to change the settings of the prompt operands.
2. Use the FEATURES ENABLE/DISABLE PROMPT configuration file statement to initialize the settings of the prompt operands at IPL time.
3. If the SET PROMPT command was not used or the FEATURES ENABLE PROMPT statement was not specified in the configuration file, the operator will not be prompted for start up information after a system restart or a shutdown REIPL.

### Responses

Response 1:

To query the settings of the prompt after\_restart operand enter the following:

```

q prompt after_restart
The system will not prompt for startup information after restart
Ready;
  
```

To query the settings of the prompt after\_shutdown\_reipl operand enter the following:

```

q prompt after_shutdown_reipl
The system will not prompt for startup information after a
shutdown reipl
Ready;
  
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6704E Missing token at end of line

## QUERY PROMPT

- HCP6706E Invalid string - *string*

# QUERY PSWTRANS

► Query — PSWTRANS ◄

## Authorization

Privilege Class: G

## Purpose

Use QUERY PSWTRANS to determine if the PSW option will be accepted on the DISPLAY, DUMP and STORE commands to display or manipulate 64-bit ESA/390 PSWs when the virtual machine is in z/Architecture mode.

## Usage Notes

1. To change the PSWTRANS option, enter the SET PSWTRANS command.

## Responses

Response 1:

```
PSWTRANS {ON      }
          {OFF     }
          {ALL     }
          {DISPLAY }
          {STORE  }
```

### ON

indicates that the PSW option is allowed for the DISPLAY, DUMP or STORE commands when the virtual machine is in z/Architecture mode.

### OFF

indicates that the PSW option is not allowed for the DISPLAY, DUMP or STORE commands when the virtual machine is in z/Architecture mode.

### ALL

indicates that the PSW option will be accepted for the DISPLAY, DUMP and STORE commands to reference or change PSW data in the 64-bit ESA/390 format when the virtual machine is in z/Architecture mode (64-bit PSW displayed and 64-bits accepted on STORE).

### DISPLAY

indicates that the PSW option will be accepted for the DISPLAY and DUMP commands (but not STORE) to reference PSW data in the 64-bit ESA/390 format when the virtual machine is in z/Architecture mode.

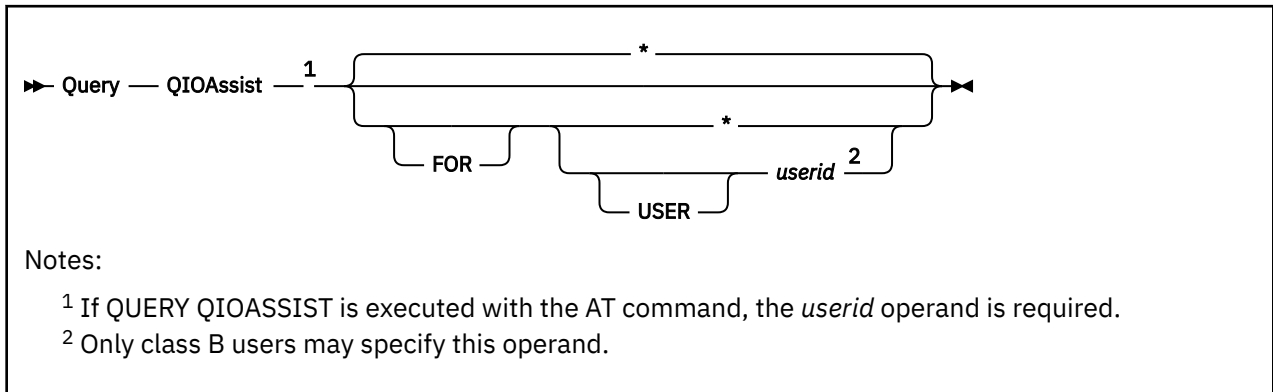
### STORE

indicates that the PSW option will be accepted for the STORE command (but not DISPLAY or DUMP) to change PSW data in the 64-bit ESA/390 format when the virtual machine is in z/Architecture mode.

## Messages

- HCP003E Invalid operand - *operand*

## QUERY QIOASSIST



### Authorization

Privilege Class: B, G

### Purpose

Use QUERY QIOASSIST to determine the current status of the queue-I/O assist for a virtual machine. The setting for queue-I/O assist is controlled by the SET QIOASSIST command.

### Operands

#### USER *userid*

\*

specifies an individual user for which the queue-I/O assist status is to be reported.

An asterisk (\*), which is the default, specifies the virtual machine (user ID) from which the command is issued.

### Usage Notes

1. When the setting for all users is OFF, eligibility for queue-I/O assist is disabled for all virtual machines, including any new users that may logon, regardless of the individual setting.

When the setting for all users is ON, then the individual user setting determines whether queue-I/O assist is allowed or disallowed for the requested user ID.

### Responses

The response displays the setting for all users, followed, if appropriate, by the individual user setting and status, which is one of the following:

#### ACTIVE

the specified *userid* is currently enabled to use the queue-I/O assist for devices that support the queued-I/O protocols.

#### PENDING ACTIVE

the specified *userid* is in the process of being enabled to use the queue-I/O assist.

#### PENDING INACTIVE

the specified *userid* is in the process of being disabled from using the queue-I/O assist.

#### INACTIVE

the specified *userid* is currently not using the queue-I/O assist.

#### NOT AVAILABLE

the queue-I/O assist is not available to the virtual machine. See [“SET QIOASSIST” on page 1623](#).

**USER DISABLED**

the virtual machine was made ineligible to use the queue-I/O assist by the SET QIOASSIST OFF command for the userid.

**CP DISABLED**

the virtual machine cannot use the queue-I/O assist because of special conditions in the virtual machine environment. For example, if the user enabled the tracing of I/O interruptions via the TRACE I/O or TRACE AIF CP commands, the queue-I/O assist is disabled in order to trace the events.

**Messages**

- HCP003E Invalid operand - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command

## QUERY QUICKDSP

---

►► Query — QUICKDSP — *userid* ◄◄

### Authorization

Privilege Class: A, E

### Purpose

Use QUERY QUICKDSP to display the QUICKDSP attribute (on or off).

### Operands

#### *userid*

is the virtual machine whose QUICKDSP setting you want displayed.

### Responses

Response 1:

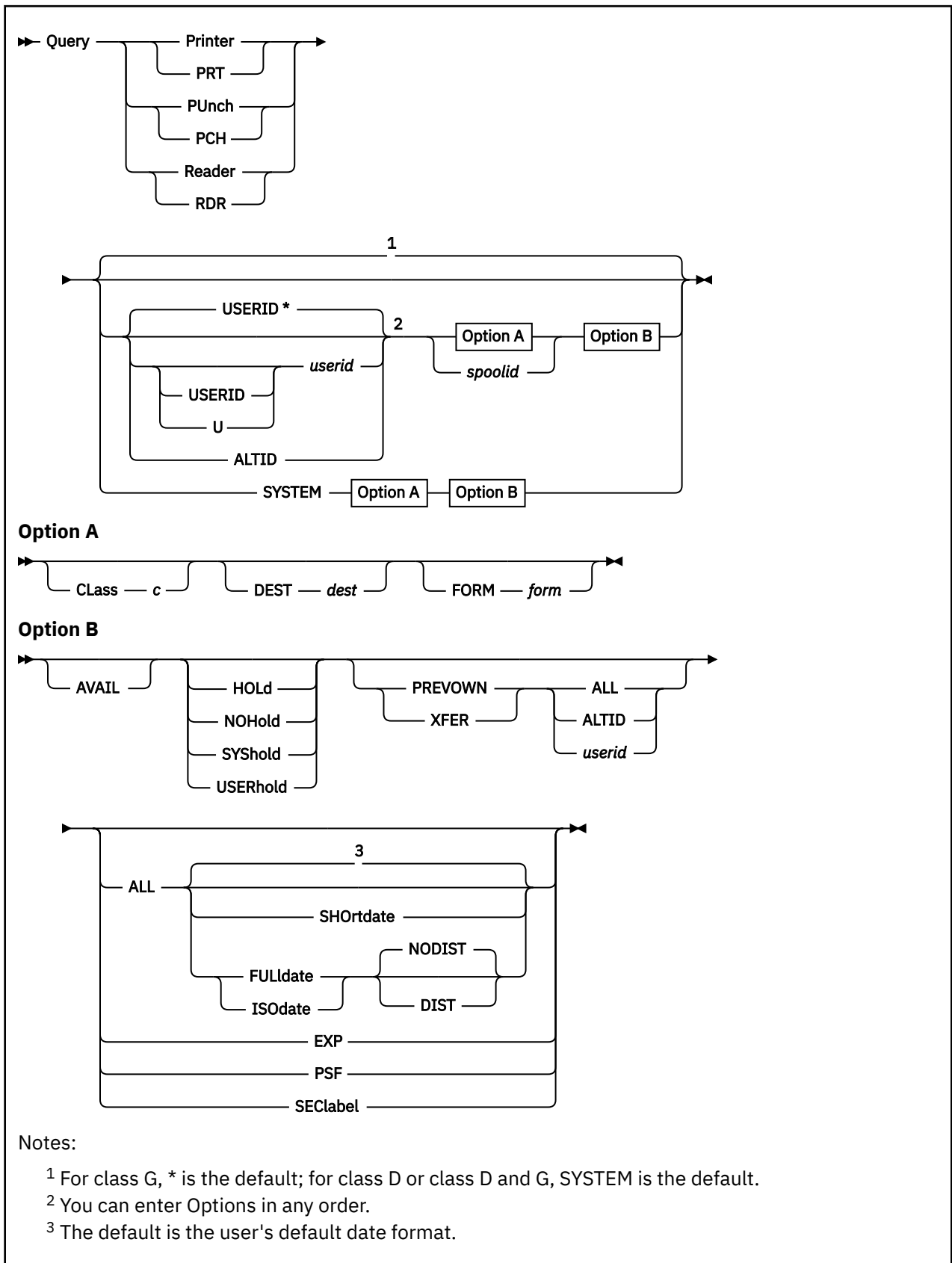
```
USER userid: QUICKDSP={ON }  
                {OFF }
```

is returned to the QUERY QUICKDSP command issuer.

### Messages

- HCP007E Invalid *userid* - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on

# QUERY READER/PRINTER/PUNCH



## Authorization

Privilege Class: D, G

## Purpose

Use QUERY READER/PRINTER/PUNCH to display detailed information about one or more spool files of the specified type. The display includes files that are:

- Open on virtual devices
- Closed and on the device queue
- Being processed on real printers, punches, or readers.

Spool files are displayed in the order in which they are processed.

Class G users can obtain information about reader, printer, or punch files that they own, that they created, or that are owned by their alternate user ID.

Class D users can obtain information about all spool files in the system, or spool files belonging to any user on the system.

## Operands

### Printer

**PRT**

### PUnch

**PCH**

### Reader

**RDR**

displays the following information about your reader, printer, or punch files:

- The *userid* of the file creator
- The spool file identification
- The spool file's class and type of device where originated
- The number of logical records in the file
- The number of copies specified for the file
- The file HOLD, KEEP, and MSG status
- The form name.

### USERID

**U**

indicates that the next parameter is a userid. Class G users can specify only an asterisk (\*) following USERID or U. U is the only abbreviation allowed for USERID.

**\***

### *userid*

**ALTID**

identifies the user whose spool files are to be queried. You can specify an asterisk (\*) to query your own files; an asterisk (\*) is the default for the class G user. You can specify ALTID to query files belonging to your alternate user ID. Class G users cannot specify *userid*.

Class D users can obtain information about reader, printer, or punch spool files that are owned by any user of the system by specifying that user's *userid*. If class D users want to obtain a display about their own spool files, they must enter their own *userid* or an asterisk (\*) on the command line. A class D user issuing the QUERY command for a one-to-four digit, all numeric *userid*, will cause unpredictable results because of the one-to-four digit, all numeric spoolid parameter.



**SYSTEM**

allows only the class D user to obtain information about all files in the system, regardless of ownership. This system is not allowed with the *spoolid* option. SYSTEM is the default for the class D user.

***spoolid***

is the system-assigned number of the spool file that is to be queried. This option is not valid with the SYSTEM keyword.

**Options****Class *c***

designates an existing spool file class. The operand *c* is a 1-character alphanumeric field that may be from A to Z, or 0 to 9.

**DEST *dest***

indicates that you want to obtain information for files with the specified destination value; *dest* is a 1- to 8-character value.

**FORM *form***

indicates that you want to obtain spool file information for files with the specified form; *form* is one to eight characters long. For the class G user, *form* is the user form name. For the class D operator, *form* is the user form name when you specify asterisk (\*). Otherwise, *form* is the operator form number.

**AVAIL**

displays only available files, not those files which are open or in use.

**HOLD****NOHold****SYShold****USERhold**

identifies the status of the files to be queried.

- Specify HOLD to query files that are in user or system hold. A file that is in HOLD status cannot be processed by the spooling device. If you are a class G user, you can change a file's user hold status by using the CHANGE command; if you are a class D user, you can also change the system hold status by using the CHANGE command.
- Specify NOHOLD to query files that are not in user or system HOLD.
- Specify SYSHOLD to query files that only have system HOLD in effect.
- Specify USERHOLD to query files that only have user HOLD in effect.

**ALL**

displays additional information for spool files. In addition to the information normally displayed for the reader, printer, or punch files, the following is also displayed:

- The date and time the file was created
- The file name and file type of file (if any)
- The distribution code of the file. If FULLDATE or ISODATE is specified, the distribution code is displayed only if DIST is also specified. See the DIST/NODIST option for details.

**Note:** The form name is not displayed when you use the ALL option.

**EXP**

displays expanded information for spool files.

The information displayed is:

- The user form name and operator form number
- The forms control buffer (FCB)
- The name and number of the copy modification module
- The flash name

- The number of copies to be flashed (flash count)
- The position of the 3800 LOAD CCWs within the spool file
- The name of the character set or sets
- The size of the file. This is the number of 4096-byte blocks in the file.

TBL is accepted as a substitute for the EXP option.

### **PSF**

displays additional information for spool files. This additional information is returned under the following headings:

### **DEST**

identifies which printers are to process the file.

### **CONVERSION**

indicates whether PSF has done preliminary processing on the file.

### **SPECIAL**

indicates whether the file contains X'5A' CCWs inserted by CP. The presence of X'5A' CCWs indicates that PSF control information has been added to the file.

### **PURGE**

indicates whether the file has been purged from CP and is waiting for PSF to recognize this state.

### **SEClabel**

displays the security label associated with each spool file.

### **PREVOWN *userid*|ALL|ALTID**

displays information about files previously owned by the specified user ID or ALTID and owned by the issuing user ID. Class D users may use the SYSTEM option to display previous owner information about any files in the system. PREVOWN *userid* displays information about spool files transferred by the specified user ID or ALTID and owned by the query issuer. Files transferred back to the generator are not displayed. PREVOWN ALTID displays information about files generated by the ALTID owned by the issuing user ID. PREVOWN ALL displays information about files not generated by the issuing user ID or ALTID that are owned by the issuing user ID.

### **XFER *userid*|ALL|ALTID**

displays information about spool files generated by the specified user ID or ALTID and owned by another user ID. Class D users may use the SYSTEM option to display information about any files in the system owned by a user ID other than the originating user ID. XFER *userid2* displays information about spool files generated by the specified user ID or ALTID and owned by the user specified as *userid2*. Files generated and owned by the same user are not displayed. XFER ALTID displays information about files generated by the specified *userid* and owned by ALTID. XFER ALL displays information about files generated by the specified *userid* or ALTID and owned by another *userid*.

### **SHOrtdate**

specifies that dates in the response to this command will be displayed in *mm/dd* format, where *mm* is the month and *dd* is the day of the month.

### **FULLdate**

specifies that dates in the response to this command will be displayed in *mm/dd/yyyy* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

### **ISOdate**

specifies that dates in the response to this command will be displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

### **NODIST**

### **DIST**

Specifies whether or not to display a distribution code when FULLDATE or ISODATE has been specified. The default is NODIST so each output record fits within an 80-character buffer.

## **Usage Notes**

1. TBL is an acceptable substitute for the EXP option. The response has the same wording.

2. If the *userid* option is used with the XFER option, the *userid* option is treated as the origin ID of the files. The result is information about the files generated by the first user and which are owned by the user specified after the XFER option.
 

If a Class D user enters QUERY PRT USER1, the user receives information about the printer files that USER1 owns.

If a Class D user enters QUERY PRT XFER USER2, the user receives information about the printer files that USER2 owns but did not originate.

If a Class D user enters QUERY PRT USER1 XFER USER2, the user receives information about the printer files generated by USER1 but owned by USER2.
3. For Class D Users:
  - The *userid* and SYSTEM options are permitted for class D users only.
  - If you use a one-to-four digit, all-numeric *userid*, the QUERY command produces unpredictable results because it also has a one-to-four digit, all-numeric *spoolid* parameter.
  - If you specify *spoolid*, you must also specify *userid* or an asterisk (\*).
4. If DEST is not specified on a command that can select files by DEST, the destination value of the file is ignored. The AVAIL option limits the files displayed to include only those files which are currently available to the virtual machine. When the virtual machine issues a read to the virtual card reader and a reader file is not already open, the first file which is available is selected. It is possible for a file not currently available to be on the chain ahead of another which is available (that is, CP is checkpointing the file). And it is possible that this file can become available between the time that this QUERY command is issued and the time that the virtual machine actually issues a read. If it is important that a particular file be selected on the read, then that file should be placed at the beginning of the queue with the ORDER command before the read is issued. An alternative for use in a program is Diagnose X'14' subcode X'2C'. This function allows a program to choose a particular reader file, order it, and read the first record all in one operation.
5. The SECLABEL option is valid only when an external security manager (ESM) is installed and security label checking is enabled.
6. When security label checking is enabled, the ESM is called to verify the user's access to each file. If the user is denied access to a specific file, some of the information about that file must be hidden from the user. The only fields of the response that will contain information are ORIGINID, FILE, CLASS, HOLD, DATE and TIME. All other fields contain asterisks. If a security label is not assigned to the file, the SECLABEL field contains the word NONE.
7. The default date format for certain CP and CMS commands can be set on a system-wide basis and also for the individual user. The system-wide default date format is set with the SYSTEM\_DATEFORMAT system configuration statement. The user's default date format is set with the DATEFORMAT directory statement. The system-wide default and the user's default can also be set with the CP SET DATEFORMAT command. The user's default date format defaults to the system-wide default. The system-wide and user settings can be queried with the CP QUERY DATEFORMAT command. The hierarchy of possible date format settings for the QUERY RDR/PRT/PUN command, from highest priority to lowest, is:
  - QUERY RDR/PRT/PUN command option
  - User default
  - System-wide default
8. The separate commands QUERY VIRTUAL READER, QUERY VIRTUAL PRINTER, and QUERY VIRTUAL PUNCH do not facilitate per-*userid* versions of this data request. They display the status of the virtual devices themselves, whereas this command details information about one or more spool files of the specified type.

## Responses

The column heads and variables used in the following responses are displayed at the end of this section under [Explanation of Response Contents](#).

## QUERY READER/PRINTER/PUNCH

Response 1:

The following response is issued when you specify QUERY RDR/PRT/PUN without the ALL option:

*For the Class G user*

```
ORIGINID FILE CLASS RECORDS  CPY HOLD FORM      DEST KEEP MSG
userid  spid a typ nnnnnnnn  nnn NONE formname dest ON  ON
                    *nnn USER          OFF OFF
                    SYS
                    USYS
```

*For the Class D user*

```
OWNERID FILE CLASS RECORDS  CPY HOLD USERFORM OPERFORM DEST KEEP MSG
userid  spid a typ nnnnnnnn  nnn NONE userform operform dest ON  ON
                    *nnn USER          OFF OFF
                    SYS
                    USYS
```

Response 2:

The following response is issued when you specify QUERY RDR/PRT/PUN with the ALL and SHORTDATE option, or if the user's default date format setting is SHORTDATE:

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME      NAME      TYPE      DIST
userid  spid c typ nnnnnnnn  nnn NONE mm/dd hh:mm:ss filename
                    *nnn USER
                    SYS
                    USYS

filetype distcode
```

Response 3:

If you use the EXP option, you receive the following response:

```
ORIGINID FILE CLASS RECORDS  FLASH FCB MDFY  FLSHC LOAD CHARS      SIZE
userid  spid c typ nnnnnnnn  name fcb mod n ccc NO name name name name size
BEG
ANY
```

**Note:** If the size is greater than 9999, then sssK will be shown, and if the size is greater than 999499, then sssM is shown. "K" means 1000 and "M" means 1,000,000.

Response 4:

If the file is open (being created or read) on a virtual device, then the OPEN status and the virtual device number (*vdev*) are displayed in place of the DATE and TIME fields.

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME NAME
userid  spid c typ nnnnnnnn  nnn NONE OPEN- vdev filename
                    *nnn USER
                    SYS
                    USYS

TYPE      DIST
filetype distcode
```

Response 5:

If a closed spool file is in the process of being printed or punched on a real output device, the PRT or PUN status and the number of records remaining to be printed in the current copy are displayed in the DATE and TIME fields.

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME NAME
userid  spid c typ nnnnnnnn  nnn NONE PRT-  nnnn filename
                    *nnn USER PUN-  nnnK
                    SYS      nnnM
                    USYS

TYPE      DIST
filetype distcode
```

**Note:** If the number of records remaining to be printed is greater than 9999, *nnnK* is shown, and if the number of records remaining is greater than 999499 then *nnnM* is shown. "K" means 1000 and "M" means 1,000,000.

Response 6:

If a spool file is open (being read) on a real card reader, the OPEN status and the real device number (*rdev*) are displayed in place of the DATE and TIME fields.

```

ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME NAME
userid  spid c typ nnnnnnnn  nnn NONE OPEN- rdev filename
                    *nnn USER
                    SYS
                    USYS

TYPE      DIST
filetype distcode
    
```

Response 7:

The following response is issued when you specify QUERY RDR/PRT/PUN with the PSF option:

```

ORIGINID FILE CLASS RECORDS  FORM      DEST CONVERSION SPECIAL PURGE
userid  spid c typ nnnnnnnn  formname dest constat  YES   YES
                    NO      NO
    
```

Response 8:

The following response issued by QUERY PRINT for a file that has been purged.

```

ORIGINID FILE CLASS RECORDS  CPY HOLD **PURGED** NAME
userid  spid c typ nnnnnnnn  nnn NONE          filename
                    *nnn USER
                    SYS
                    USYS

TYPE      DIST
filetype distcode
    
```

The response indicates that PURGE processing is not complete for the file. This is a temporary state. If a printer spool file remains in this state for an extended period of time, it may indicate that the file has been selected for printing, but that intervention is required on the printer, or it may indicate that there is a problem with the printer server virtual machine.

Response 9: (QUERY PRINTER only)

When CP manages a real 3800 system printer, it maintains a delayed purge queue for each 3800 system printer. If the files have already been printed, but are being held in this delayed purge queue, the text **\*\*PRINTED\*\*** will be displayed in the column that usually contains the date and time.

```

ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME  NAME
userid  spid c typ nnnnnnnn  nnn SYS  **PRINTED** filename

TYPE      DIST
filetype distcode
    
```

The delayed purge queue for each printer is limited, so this file should eventually be purged by the system when it gets 'pushed' off the queue by subsequent print jobs. A file marked **\*\*PRINTED\*\*** cannot be changed, transferred, ordered, tagged, or dumped by SPXTAPE. It can be deleted by the PURGE command, which can be issued by the operator, or by the owner of the file.

These files may become detached from the delayed purge queue by events like an abnormal system shutdown. When this happens, the file will not be automatically deleted by the system. It should be deleted by the operator or the owner of the file.

Response 10:

If no files of the specified type exist, one of the following responses is given, corresponding to the specified file type:

```
NO RDR FILES
```

- or -

NO PRT FILES

- or -

NO PUN FILES

Response 11:

In an SSI cluster, the response for QUERY RDR/PRT/PUN ALL shows whether a file is unavailable for processing.

LOCK- *sysname* means the file is locked by (another) system, *sysname*. When a spool file is requested and the file is unavailable to the inquiring system because it is in use by another system, a status of LOCK appears under DATE, and the system that is using the spool file appears under TIME. For example:

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME  NAME TYPE DIST
userid   spid c typ nnnnnnnn  nnn NONE LOCK-  sysname fnft distcode
```

COPY- *sysname* means this system's SPFBK is a copy, and the lock cannot be obtained because communication is not available to the system that owns the original. When the requested spool file is a copy and the file cannot be processed because communications with the owning system are impossible, a status of COPY appears under DATE, and the system that owns the spool file appears under TIME. For example,

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME  NAME TYPE DIST
userid   spid c typ nnnnnnnn  nnn NONE COPY-  sysname fnft distcode
```

In these circumstances, a reader file may be read using the PEEK or RECEIVE commands, but it cannot be manipulated by the CHANGE or TRANSFER commands.

In all other cases, the response to this command in an SSI cluster is the same as the response when not in an SSI cluster.

Response 12:

The following response is issued when you specify QUERY RDR/PRT/PUN with the SECLABEL option:

*For a class G user*

```
ORIGINID FILE CLASS RECORDS  CPY HOLD FORM      DEST      SECLABEL
userid   spid a typ nnnnnnnn  nnn NONE formname dest  ssssssss
                *nnn  USER
                SYS
                USYS
```

*For a class D user*

```
OWNERID  FILE CLASS RECORDS  CPY HOLD USERFORM
userid   spid a typ nnnnnnnn  nnn NONE userform
                *nnn  USER
                SYS
                USYS

OPERFORM DEST      SECLABEL
operform dest  ssssssss
```

Response 13:

When security label checking is enabled, some fields must be hidden from the user if access to the file is denied. This example shows the response that is issued when you specified QUERY RDR/PRT/PUN with the SECLABEL option, and you were denied access to the file:

*For a class G user*

```
ORIGINID FILE CLASS RECORDS  CPY HOLD FORM      DEST      SECLABEL
userid   spid a typ *****  *** NONE *****  ****  *****
                ****  USER
```

```
SYS
USYS
```

Response 14:

If the spool file that was printing has been suspended, the number of records left to be printed is displayed in place of the DATE and TIME fields.

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME NAME
userid  spid c typ nnnnnnnn  nnn nnnn nnnn  rdev filename
                                *nnn USER
                                SYS
                                USYS

TYPE      DIST
filetype distcode
```

Response 15:

The following response is issued when you specify QUERY RDR/PRT/PUN with the ALL, FULLDATE, and NODIST options:

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE      TIME  NAME
userid  spid c typ nnnnnnnn  nnn NONE mm/dd/yyyy hh:mm:ss
filename
                                *nnn USER
                                SYS
                                USYS

TYPE
filetype
```

Response 16:

The following response is issued when you specify QUERY RDR/PRT/PUN with the ALL, ISODATE, and DIST options:

```
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE      TIME  NAME
userid  spid c typ nnnnnnnn  nnn NONE yyyy-mm-dd hh:mm:ss
filename
                                *nnn USER
                                SYS
                                USYS

TYPE      DIST
filetype distcode
```

Explanation of Response Contents:

**CHARS name**

is the name of the character set or sets used to generate output if printed on a 3800 printer.

**CONVERSION convstat**

is the conversion status of the file, which can be any of the following:

Status	Meaning
<b>ACTIVE</b>	The print server is converting the file to a CMS file containing the actual data and the print controls used to print the file.
<b>COMPLETE</b>	The file is converted.
<b>NOTCONV</b>	The file is not yet been converted.

**DEST dest**

specifies the destination value assigned to the spool file.

**distcode**

is the distribution code of the file.

**FCB fcb**

is the name of the FCB module used in the vertical formatting of a page.

**filename**

is the file name assigned to the file (if any).

**filetype**

is the file type assigned to the file (if any).

**FLASH name**

is the name of the forms overlay frame to be superimposed on the output if printed on a 3800 printer.

**FLSHC ccc**

is the number of copies that are printed while the forms overlay frame is in place if printed on a 3800 printer.

**FORM name**

is the name of the paper type as understood by the user (displayed for a class G user).

**hh:mm:ss**

is the time when the file was created in hours, minutes, and seconds.

**KEEP ON****KEEP OFF**

indicates whether the file is retained after being processed. ON indicates that KEEP is in effect and the file is retained after processing, and OFF indicates that KEEP is not in effect.

**LOAD NO****LOAD BEG****LOAD ANY**

is where the 3800 LOAD CCWs are positioned within the spool file—not at all (NO), at the beginning (BEG), or interspersed throughout the spool file (ANY).

**MDFY mod**

is the name of the copy modification module used to alter output text if printed on a 3800 printer.

**mm/dd****mm/dd/yyyy****yyyy-mm-dd**

is the date the file was created. The format of the date is determined by the SHORTDATE, FULLDATE, or ISODATE option. If no date format option is specified, the user's default date format setting is used.

**MSG ON****MSG OFF**

indicates whether messages about the file printing or punching are displayed. ON indicates these messages are displayed; OFF indicates they are not.

**n**

is the copy modification character set number.

**nnn****\*nnn**

is the number of copies requested for the spool file. This is shown as a 3-digit number (*nnn*). The asterisk (\*) indicates that the 3800 printer should print each page from the spooled output files *nnn* times before going on to the next page.

**nnnnnnnn**

is the number of logical records contained in the file.

**NONE****USER****SYS****USYS**

represents the file's HOLD status. NONE indicates there are no holds on the file, USER indicates the file is in user hold status, SYS indicates the file is in system hold, and USYS indicates the file is in both user and system hold.

**operform**

is the name of the paper type as understood by the operator (displayed for a class D user).



**ORIGINID****OWNERID**

ORIGINID heading the title line for the spool file data is altered to OWNERID when *userid* or the XFER operand is used, or when a class D user is defaulting to, or specifying, SYSTEM. In that event, OWNERID represents the owner of the file.

**PURGE (YES/NO)**

YES indicates that a file has been converted and subsequently changed, transferred, or purged. A new and unconverted file was created to reflect the altered attributes while retaining the original file data.

When the print server selects a file with PURGE=YES, it cancels further processing and deletes the old file.

**SECLABEL ssssssss**

identifies the security label (SECLABEL) associated with each file. The SECLABEL value is displayed as a 1- to 8-character alphanumeric value.

**SIZE size**

is the number of 4 KB DASD blocks allocated for data for the file.

**SPECIAL (YES/NO)**

indicates whether the file contains any records with X'5A' CCWs. The presence of X'5A' CCWs indicates that CP has added PSF control information to the file.

**spid**

is a system-assigned number used by the system to identify the file.

**typ**

identifies the device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP or SYS.

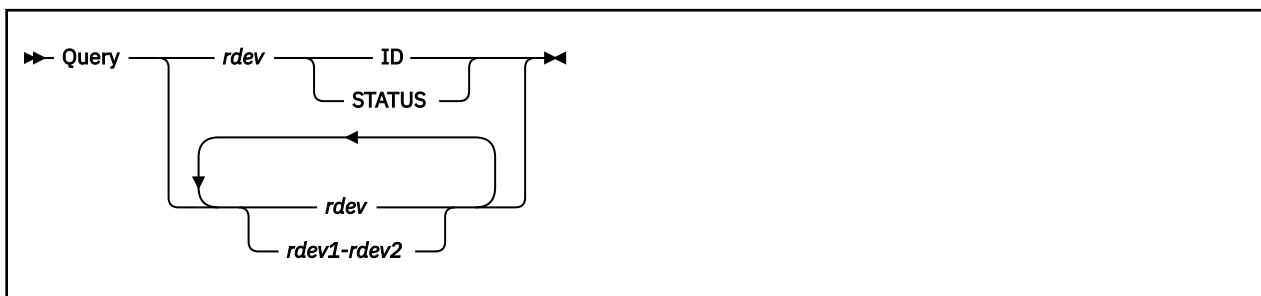
**userform**

is the name of the paper type as understood by the user (displayed for a class D user).

**Messages**

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP849E A *userid* must be specified with the spoolid
- HCP1558E Command processing cannot complete.

## QUERY (Real Device)



### Authorization

Privilege Class: B

### Purpose

Use QUERY (Real Device) to display the status of real devices.

### Operands

#### ID

displays the device and control unit information from the sense ID data for a specified device address if they are known. It also displays the device equivalency ID (EQID) if one exists for the device.

#### *rdev*

#### *rdev1-rdev2*

displays the kind of real device at a specified device address. The status presented indicates whether the device is operational, has an equipment check, has a channel error, or is offline. You can enter a single *rdev*, a series (*rdev rdev*), or a range (*rdev1-rdev2*). Each *rdev* in the active configuration can be entered as a 4-digit hexadecimal device number between X'0000' and X'FFFF'. For an *rdev* in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is defined. A range of devices cannot span subchannel sets.

#### STATUS

displays the status of a real DASD. The status indicates whether the device is operational, has an equipment check, has a channel error, or is offline.

### Usage Notes

1. If you are authorized for some options of this command but not for others and you enter an unauthorized option, CP interprets your command in the formats for which you are authorized. This might cause the command to run in an unexpected fashion, or reflect an unexpected message.
2. If the ID option is specified, but sense ID data is not available, the information is displayed as a dash (-).

### Responses

Response with ID:

```
type rdev dddd-mm CU: cccc-nn
    rdev EQID: qqqqqqqq
```

**type**

is the device type

**rdev**

is the real device number

**dddd**

is the device identifier

**mm**

is the device model number in hexadecimal

**cccc**

is the control unit identifier

**nn**

is the control unit model number in hexadecimal

**qqqqqqqq**

is the 1–8 character user-defined device equivalency ID (EQID) or the 50 character system-generated EQID.

**Note:** If the device identifier is unknown, the *dddd* and *mm* fields are blank. If the control unit identifier is unknown, the *cccc* and *nn* fields are blank. If the device has no EQID, the *qqqqqqqq* field is blank.

Responses with STATUS:

**Note:** In all of the following responses, *rdev* is the real device number for which the user refers to the device.

When the device is ready and available, you see a response like this:

```
DEVICE rdev OPERATIONAL
```

When the device is either disabled or not connected to the channel subsystem, you see a response like this:

```
DEVICE rdev NOT OPERATIONAL
```

When the device sense data indicates an equipment check, you see a response like this:

```
DEVICE rdev EQUIPMENT CHECK
```

When the device has indicated it has hardware problems, you see a response like this:

```
rdev CHANNEL ERROR
```

When the device is unavailable (boxed), you see a response like this:

```
DASD rdev BOXED
```

When the device is unavailable to CP, you see a response like this:

```
DASD rdev OFFLINE
```

When the device is in a state-change-pending condition, during which it cannot service I/O requests, you see a response like this:

```
DASD rdev STATE CHANGE IS PENDING
```

When the subchannel for the device has been varied offline you see a response like this:

```
DASD rdev SUBCHOFF
```

When the device has been taken offline for maintenance, you see a response like this:

## QUERY (Real Device)

```
DASD rdev QUIESCED
```

Response for Devices Needing Intervention:

If the system is using a device that is in an intervention-required state when you enter QUERY (Real Device), you receive a response that looks like this:

```
DASD 1801 has intervention required.
```

DASD is the real device type. It can be replaced by one of the following, depending on the type of device at the specified address:

<b>Type</b>	<b>Meaning</b>
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
LINE	Communication line
CTCA	Channel-to-channel adapter
FCP	SCSI Device Adapter
OSA	Open Systems Adapter
CTLR	3745 Communications Controller
MSC	Mass Storage Controller
SWCH	Dynamic Switching device
DEV	Any other device.
KBD	Keyboard

1801 is the real device number of the device that needs the operator's intervention.

**Note:** This response is preceded by one of the following.

Responses from Real Channel-to-Channel Adapters:

If there are channel-to-channel adapters attached to the system at the addresses specified, you see something like this:

```
q 480-486
DEV 0480 OFFLINE , DEV 0481 OFFLINE , CTCA 0482 FREE
CTCA 0483 ATTACHED TO RSCS 0483
CTCA 0484 FREE
CTCA 0485 ATTACHED TO PVM 0484
CTCA 0486 ATTACHED TO VTAM 0486
Ready;
```

The first number is the real device number of the channel-to-channel adapter. The ATTACHED operand might also read BOX/ATTC, meaning that although the device is attached and dedicated to a user, it is unavailable. PVM, RSCS, VTAM are user IDs to which the CTCAs are dedicated. The last number is the virtual device number which each virtual machine uses to refer to a CTCA. Any free, boxed, or offline devices within the device range are also included in the response.

Specifying the ID option for a single device may display the following sense ID and control unit information:

```
q 7e0 id
CTCA 7E0 - CU: 3088-08
Ready;
```

**Note:** Use the QUERY CTCA command to restrict responses to include only those CTCA devices in a specific state, such as free, boxed, or offline.

Responses from Real DASD:

You see responses like the following if real DASDs are attached to the system for access as user minidisks or for system paging, spooling, or temporary disk space.

```
q 280-285
DASD 0280 CP SYSTEM XAUSR1 49
DASD 0281 CP SYSTEM XAUSR2 24
DASD 0282 CP SYSTEM XAUSR3 49
DASD 0283 CP SYSTEM XAUSR4 34
DASD 0284 CP SYSTEM XAUSR5 42
DASD 0285 CP SYSTEM XAUSR6 20
Ready;
```

The first number listed is the real device number of the DASD. SYSTEM indicates the device is allocated to the system for use as users' minidisks. OWNED indicates the device is used by the system for paging and spooling activity. XAUSR1 and the rest are volume labels of the devices. The last number indicates the number of links to minidisks contained on the volume.

The following response is also displayed if real DASDs are attached to the system for access as user minidisks or for system paging, spooling, or temporary disk space. This response is longer than the response above because the device may be reserved (shared option is on), it may have a reserve pending, or it is currently reserved.

```
q 181a-181c
DASD 181A CP SYSTEM ESAP2C SHARED
DASD 181B CP OWNED ESAP2F RESERVE PENDING FOR USER DAVID
DASD 181C CP OWNED ESAP30 RESERVED BY SYSTEM
Ready;
```

If a DASD at the specified address is dedicated to a user, you will see a message like this:

```
q 820
DASD 0820 ATTACHED TO OLIVER 01FC R/O XAUSR4
Ready;
```

This means that the DASD at the real device number 0820 is attached to OLIVER's virtual machine as the virtual device 01FC. The DASD's volume identifier is XAUSR4. R/O means that the user OLIVER has read-only access to the DASD. R/O might also read R/W, which would mean that the user had read-write access. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

If the device was defined by the DEVNO option on the MDISK statement, you see a response like this:

```
QUERY F8A3
DASD F8A3 CP SYSTEM DEVNO 13
```

The first number listed is the real device number of the DASD. SYSTEM indicates the device is allocated for use as a minidisk. DEVNO indicates that this device was defined by the DEVNO option on the MDISK statement. The last number indicates the number of links to the full-pack minidisk.

There must be at least one link, either by the owner of the disk or another user, to get the DEVNO response. One cannot link to a DEVNO mdisk attached to the system, it must be a free device. Even when linked, a DEVNO device is not attached to the system.

The volser for a DEVNO device is available from QUERY SYSTEM rdev.

Responses from Real Graphic Display Devices:

## QUERY (Real Device)

If a real graphic display at the specified address is dedicated to a virtual machine, you see something like this:

```
q 0bcd
GRAF 0BCD ATTACHED TO OLIVER 0010
Ready;
```

This message means the graphic display device at the real device number 0BCD is attached to OLIVER's virtual machine as the virtual device 0010. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

If a graphic display device has a user logged on or dialed through it, you see something like this:

```
q 08e0
GRAF 08E0 LOGON AS SVCDIR 0009
Ready;
```

This response tells you that the user SVCDIR is logged on through the graphic display device at 08E0. The number 0009 is SVCDIR's virtual console device number.

If there are free graphic display devices at the specified addresses, you see something like this:

```
q 08e1-08e7
GRAF 08E1 ENABLED, GRAF 08E2 ENABLED, GRAF 08E3 ENABLED,
GRAF 08E4 LOGON AS OPERATOR 0009
GRAF 08E5 OFFLINE, GRAF 08E6 ENABLED, PRT 08E7 ENABLED
Ready;
```

**Note:** Use the QUERY GRAF command to restrict responses to include only those GRAF devices in a specific state, such as free, boxed, or offline.

Responses from Real Communication Lines:

If a real communication line is dedicated to a user, you see something like this:

```
q f0
LINE 00F0 ATTACHED TO OLIVER 0010
Ready;
```

This response means that the communication line with the real device number 00F0 is attached to OLIVER's virtual machine as the virtual device 0010. ATTACHED may be replaced with BOX/ATTC, which means that although the device is dedicated to a user, it is unavailable (usually because of a hardware condition).

If there are communication lines that have users logged on at them or dialed through them, you see something like this:

```
q fa fc
LINE 00FA LOGON AS DAVID 0009
LINE 00FB DIALED TO PVM 00F9
LINE 00FC FREE
Ready;
```

This response tells you that the user DAVID is logged on through the communication line at 00FA. The number 0009 is the virtual device number of the virtual console where DAVID is logged on. The second line tells you that the terminal connected to communications line 00FB is dialed to PVM's virtual device number 00F9.

**Note:** Use the QUERY LINE command to restrict responses to include only those LINE devices in a specific state, such as free, boxed, or offline.

Responses from FCP Adapter Devices:

If there are FCP devices attached to the system at the addresses specified, you see something like this:

```
q 480-487
FCP 0480 ATTACHED TO LINUX1 0480 CHPID F1
FCP 0481 ATTACHED TO LINUX1 0481 CHPID F1
FCP 0482 ATTACHED TO LINUX2 0482 CHPID F1
```

```
FCP 0483 FREE, FCP 0484 FREE
FCP 0485 ATTACHED TO LINUX3 0485 CHPID F1
FCP 0486 ATTACHED TO TEST1 0486 CHPID F1
FCP 0487 FREE
Ready;
```

The first number is the real device number of the FCP adapter device. The ATTACHED operand might also read BOX/ATTC, meaning that although the device is attached and dedicated to a user, it is unavailable. LINUX1, LINUX2, LINUX3, and TEST1 are user IDs to which the FCP devices are dedicated. The next number is the virtual device number which each virtual machine uses to refer to an FCP device. The last number is the CHPID number (channel path) associated with the device number. Any free, boxed, or offline devices within the device range are also included in the response.

**Note:** Use the QUERY FCP command to restrict responses to include only those FCP devices in a specific state, such as free, boxed, or offline.

Responses from Open Systems Adapter Devices:

If there are Open Systems Adapter devices attached to the system at the addresses specified, you see something like this:

```
q 480-48B
OSA 0480 ATTACHED TO PVM 0480
OSA 0481 ATTACHED TO PVM 0481
OSA 0482 ATTACHED TO RSCS 0482
OSA 0483 ATTACHED TO RSCS 0483
OSA 0484 OFFLINE, OSA 0485 OFFLINE, OSA 0486 FREE , OSA 0487 FREE
OSA 0488 FREE OSA 0489 FREE , OSA 048A FREE , OSA 048B FREE
Ready;
```

The first number is the real device number of the Open Systems Adapter devices. The ATTACHED operand might also read BOX/ATTC, meaning that although the device is attached and dedicated to a user, it is unavailable. PVM, RSCS, TCPIP, and VMNET are user IDs to which the OSA devices are dedicated. The last number is the virtual device number which each virtual machine uses to refer to an OSA device. Any free, boxed, or offline devices within the device range are also included in the response.

Specifying the ID option for a single device may display the following sense ID and control unit information:

```
q 4a1 id
OSA 4A1 1732-05 CU: 1731-05
Ready;
```

**Note:** Use the QUERY OSA command to restrict responses to include only those OSA devices in a specific state, such as free, boxed, offline, or system.

Responses from Real Dynamic Switching Devices:

If there is a dynamic switching device dedicated to a user, you see something like this:

```
q 0cd5
SWCH 0CD5 ATTACHED TO EMILY 0012
Ready;
```

This means that the switching device at the real device number OCD5 is attached to EMILY's virtual machine as the virtual device number 0012. ATTACHED may be replaced by BOX/ATTC, which indicates the dedicated switching device is not available (usually because of a hardware condition).

**Note:** Use the QUERY SWCH command to restrict responses to include only those SWITCH devices in a specific state, such as free, boxed, or offline.

Responses from Real Magnetic Tape Drives:

If there is a tape drive dedicated to a user, you see something like this:

```
q 102
TAPE 0102 ATTACHED TO RALPH 0236 R/W
Ready;
```

## QUERY (Real Device)

This means that the magnetic tape drive at the address 0102 is attached to RALPH's virtual machine as the virtual device number 0236. R/W means that the user RALPH has read-write access to the tape drive. R/W might also read R/O, which would mean that the user had read-only access. R/W and R/O represent the software's point of view, the hardware access control status may differ.

If the tape drive is currently being used by the TRSAVE command, you see something like this:

```
q 242
TAPE 0242 ASSIGNED SYSTEM TRSAVE
Ready;
```

If a tape that is not dedicated to a user is not available (boxed), you see something like this:

```
q 0244
TAPE 0244 BOXED
Ready;
```

If the tape drive is currently being unloaded following a DETACH command, or if CP is in the process of determining whether rewind is possible, you will see something like this:

```
q 246
TAPE 0246 UNLOADING
Ready;
```

If a tape drive has been attached MULTIUSER to one or more users, you see something like this:

```
q 2100
TAPE 2100 ATTACHED TO USER1    0181 R/W  NOASSIGN MULTIUSER
TAPE 2100 ATTACHED TO USER2    0181 R/W  NOASSIGN MULTIUSER
Ready;
```

This means that the tape drive at real address 2100 is attached to USER1 and USER2 at virtual device address 0181. MULTIUSER indicates that the tape drive is shared by both users, and NOASSIGN states that the users must manage the assignment of the tape drive by using assign and unassign I/O.

**Note:** Use the QUERY TAPES command to restrict responses to include only those tape devices in a specific state, such as free, boxed, or offline.

Responses from Real Unit Record Devices:

If a unit record device is dedicated to a user, you see something like this:

```
q 000c
RDR 000C ATTACHED TO JAMES 02F0
Ready;
```

This response means that the real card reader at the device number 000C is dedicated to JAMES's virtual machine at the virtual device number 02F0. RDR might be replaced by PRT or PUN, depending on the kind of unit record device. ATTACHED might be replaced by BOX/ATTC, which indicates that the device is attached to the virtual machine but is unavailable (usually because of a hardware problem).

*Responses from Active Readers:* If a reader is assigned to the system for spooling activity and it is currently active with a spool file, you see a message like this:

```
q 000C
RDR 000C STARTED SYSTEM
RDR 000C READING PERCY FILE 0021
Ready;
```

000C is the real device number of the card reader. STARTED indicates that the device is available for spooling activity. SYSTEM means that the device is available for input with user identification cards. STARTED might be replaced by any of the following:

- DRAINING, which means that the device would be unavailable for spooling activity as soon as it finished processing the current file.
- DRAINED, which means that the device was unavailable for spooling activity; you could enter the CP START command to restart it.



- INT REQ, which means that the device was not ready and required manual intervention before spool file processing could resume.

**Note:** If STARTED were replaced by any of the above options except DRAINING, the second line of the response would not appear.

SYSTEM might be replaced by a user ID, which means that the device was available for input without user identification cards. All input decks would be assigned to the user ID specified.

The second line of the response means that the card reader is currently active with a spool file. In this example, the reader at the real device number 000C is reading a card deck and creating a file that will be the user PERCY's spool file 0021.

*Responses from Active Printers:* If there is an impact printer in use as a system spooling device, you see a response like this:

```
q 0e0e
PRT 0E0E STARTED      SYSTEM CLASS 1          SEC JOBARO
PRT 0E0E FORM STANDARD MANUAL SEP  N03800  DEFFCB  LIMIT NONE
PRT 0E0E FOLD  IMAGE IMAG4248 CHARS      FCB FCB8      LPP 068
PRT 0E0E DEST OFF      DEST - NONE - DEST - NONE - DEST - NONE -
PRT 0E0E NOAFP
Ready;
```

This response tells you the following about the impact printer at the real device number 0E0E. The first line tells you that the printer will:

- Be available for spooling activity
- Process files for any user ID
- Select only class 1 spool files for processing
- Print files with a security label (SEC) of JOBARO.

**Note:** The security label field is displayed only when an external security manager is installed and security label checking is enabled.

The second line tells you that it will:

- Select files requesting the STANDARD form
- Wait for the operator to schedule forms output
- Generate a separator page for output files
- Not accept spool files that contain 3800 load CCWs
- Use the default forms control buffer that was loaded into it
- Process spool files regardless of their length.

The third line tells you that it will:

- Translate (fold) lower case characters into upper case
- Use the system image library file IMAG4248
- Not use a character set
- Use the forms control buffer FCB8.
- Print separator pages with 68 lines per page.

The fourth line tells you that the printer will not use destination values as a criterion for selecting files.

The fifth line tells you that the printer will not process files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB). For more information about each line of the response, see the tables below.

You see a response like the following if there is a 3800 printer in use as a system spooling device and that printer is currently active with a spool file:

```
q 0f10
PRT 0F10 STARTED      SYSTEM CLASS *          SEC JOBARO
```

## QUERY (Real Device)

```
PRT 0F10 FORM STANDARD AUTO SEP ANY3800 LIMIT NONE
PRT 0F10 FLASH 0987 IMAGE IMAG3800 CHARS GF10 FCB 6 HOLD LPP 060
PRT 0F10 DEST OFF DEST - NONE - DEST - NONE - DEST - NONE -
PRT 0F10 AFP MARK
PRT 0F10 PRINTING DAVID FILE 0002 RECS 130K COPY 005 SEQ 001
Ready;
```

This response tells you the following about the printer at the real device number 0F10. The first line means that the printer will:

- Be available for spooling activity
- Process files for any user ID
- Select spool files regardless of class
- Print files with a security label (SEC) of JOBARO.

**Note:** The security label field is displayed only when an external security manager is installed and security label checking is enabled.

The second line tells you that the printer will:

- Select files requesting the STANDARD form
- Automatically schedule forms output
- Generate a separator page for output files
- Accept spool files that contain 3800 load CCWs
- Process spool files regardless of their length.

The third line tells you that the printer will:

- Use the forms overlay negative called 0987
- Use the system image library file IMAG3800
- Use the character set GF10
- Print six lines per inch
- Put spool files causing 3800 load checks into system HOLD status.
- Print separator pages with 60 lines per page.

The fourth line tells you that the printer will not use destination values as a criterion for selecting files.

The fifth line tells you that the printer will:

- Process files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB).
- Mark separator trailer pages with separator bars.

The last line tells you the following:

- The printer is active with the spool file 0002, owned by the user DAVID.
- The number of records remaining for output in the file is 130K, where K is the number of records in thousands rounded to the nearest 1000. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million.
- There are five copies remaining for output.
- The sequence for the file on the printer is 001.

The parts of the first line of the response for impact printers and 3800 printers can also be:

Category	Possible Values	Meaning
Status	STARTED	Device is available for spooling activity.
	DRAINING	Device is processing a file, but will be drained after the file has printed.
	DRAINED	Device is not available for spooling activity. Issue the CP START command to activate the device.
	INT REQ	Device is not available; it needs manual intervention before it can continue to process spool files.
	SETUP REQ	Printer is ready to print an alignment page so that the operator can align the forms properly on an impact printer. The operator must press the STOP and START buttons to print another alignment page, or enter the START command to begin printing the entire file.
	MOUNT REQ	Printer is waiting for the operator to mount a new form.
User ID	SYSTEM	Device processes files for any user ID.
	<i>userid</i>	Device processes files only for the specified user ID.
Spool file class	CLASS <i>cl</i>	Device services spool files of the specified classes, in the specified order. An output device can service as many as 8 output classes.
Security Label	<i>seclabel</i>	Printer will print only those spool files with the specified security label (SEC). The ESM-authorized operator uses the START command to change the security label for a printer.
	NONE	Printer was not assigned a security label. No files can be printed until the printer is started with a valid security label. For more information, contact your security administrator.  <b>Note:</b> NONE and <i>seclabel</i> only appear if an external security manager (ESM) is installed and security label checking is enabled.

The parts of the second line of the response can be:

Category	Possible Values	Meaning
Form	FORM <i>form</i>	Device services the specified 1- to 8-character operator form number.

Category	Possible Values	Meaning
Scheduling	MANUAL	Operator schedules forms output. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is printed, the printer stops.
	AUTO	Operator wants CP to schedule forms output. The form that is currently active is processed first. When all files with the current form have been printed, CP selects a new form for the device.
	SETUP	<p>Operator wants CP to print a page for forms alignment verification. (Only valid for impact printers.) The operator receives a message to set up the form. After setting up the form, the operator must press the STOP button, then the START button. An alignment page prints with letters replaced by X's and numbers replaced by 9's on the first page of the file.</p> <p>Each time the operator presses STOP and START after a full alignment page is printed, another alignment page prints. After aligning the forms, the operator enters the START command for the device. The printer then returns to its previous mode (AUTO or MANUAL), and the file prints in its entirety.</p> <p><b>Note:</b> If neither MANUAL, AUTO, nor SETUP is specified and the printer is currently in SETUP mode, it returns to the previous mode, AUTO or MANUAL. The default forms output is MANUAL.</p>
File separator	SEP	File separator prints between output files.
	NOSEP	File separator does not print between output files.
3800 load CCWs	NO3800	Printer does not print any spool files containing 3800 load CCWs. This option ensures a fast output queue, because load CCWs issued to 3800 printers can take a lot of time.
	BEG3800	Only spool files without 3800 load CCWs, or spool files with 3800 load CCWs only at the beginning of the file, print on this device. This allows the user an initial setup with which the entire file can be printed.
	ANY3800	Output device will process spool files regardless of whether they contain 3800 load CCWs. This option may cause a slower output queue, based on the number of 3800 load CCWs issued by the spool files.
Default FCB	DEFFCB	Printer uses the default FCB with which it was loaded. CP does not send imbedded LOAD FCB CCWs to the printer. The LOADBUF command specifies the default FCB.
	FILEFCB	<p>CP sends FCBs to the real printer. File FCBs are LOAD FCB CCWs imbedded in the spool file. This is the default when you start the printer after a cold start.</p> <p><b>Note:</b> DEFFCB and FILEFCB only appear for non-1403 impact printers.</p>

Category	Possible Values	Meaning
Length limit	LIMIT NONE	CP selects files for printing regardless of the number of records.
	LIMIT <i>nnnnnnnnnn</i>	CP only selects spool files for printing that are less than or equal to the specified number of records. The spool file record number is the number of copies multiplied by the size of the file.

The parts of the third line of the response can be:

Category	Possible Values	Meaning
Folding into upper case	FOLD	CP translates (folds) lower case characters into upper case.
	NOFOLD	CP does not translate lower case characters into upper case. <b>Note:</b> FOLD and NOFOLD only appear for impact printers.
Forms overlay negative	FLASH <i>name</i>	Printer uses the specified forms overlay negative, which is from 1 to 4 characters long. <b>Note:</b> FLASH only appears for 3800 printers.
Image library	IMAGE <i>imagelib</i>	Printer uses the specified system image library file, which contains the load images (character sets, FCBs, graphic character modifications, or copy modifications) used for printing files and separator pages.
Character set	CHARS <i>chars</i>	Printer uses the specified character set, which is from 1 to 4 characters long, when printing the separator page.
FCB	FCB <i>name</i>	For impact printers: printer uses the specified FCB when printing the file or the separator page. For 3800 printers: the FCB is the number of lines per inch. <b>Note:</b> FCB does not appear for 1403-type printers.
3800 load checks	HOLD	Printer puts spool files causing 3800 load checks in system HOLD status.
	PURGE	Printer deletes spool files causing 3800 load checks. <b>Note:</b> HOLD and PURGE only appear for 3800 printers.
Lines per page	LPP	Defines the number of lines for file separator pages on the printer.

The parts of the next-to-last line of this response can be:

Category	Possible Values	Meaning
Destination	DEST	Printer services the specified destination values. The printer can service up to 4 destination values. The first value for the device is always OFF if it was not specifically set by either a START command, or at initialization with the RDEVICE statement in the system configuration file. If it was set, the specified destination is shown. If the remaining destination values were not specified, the values are null and are shown as – NONE –. If they were specified, the specified value is shown.
AFP characteristics	AFP	Printer processes files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB).
	NOAFP	Printer does not process files with AFP characteristics.
Mark separator trailer pages	MARK	Printer marks 3800 trailer pages with separator bars.
	NOMARK	Printer does not mark 3800 trailer pages with separator bars.

**Note:** MARK and NOMARK only appear for 3800 printers.

*Responses from Active Card Punches*

If a punch is attached to a user and is currently active with a spool file, you see a message like:

```
q 1f30
PUN 1F30 STARTED SYSTEM CLASS 1
PUN 1F30 FORM STANDARD AUTO NOSEP LIMIT NONE
PUN 1F30 DEST OFF DEST - NONE - DEST - NONE - DEST - NONE -
PUN 1F30 PUNCHING JAMES FILE 0042 RECS 012K COPY 002 SEQ 001
Ready;
```

This response tells you the following about the card punch at the real device number 1F30. The first line means that the punch will:

- Be available for spooling activity
- Process files for any user ID
- Select class 1 spool files.

The second line tells you that the punch will:

- Select files requesting STANDARD form
- Automatically schedule forms output
- Not generate a separator page for output files.

The third line tells you that the punch will not use destination values as a criterion for selecting files. The parts of the first line of the response can also be:

Category	Possible Values	Meaning
Status	STARTED	Device is available for spooling activity.
	DRAINING	Device is processing a file, but will be drained when the file is finished.
	DRAINED	Device is not available for spooling activity. Issue the CP START command to activate the device.
	INT REQ	Device is not available; it needs manual intervention before it can continue to process spool files.
	SETUP REQ	Punch is ready to generate an alignment page to let the operator align the forms properly on an impact printer. The operator must press the STOP and START buttons to print another alignment page, or enter the START command to begin printing the entire file.
	MOUNT REQ	Punch is waiting for the operator to mount a new form.
User ID	SYSTEM	Device processes files for any user ID.
	<i>userid</i>	Device processes files only for the specified user ID.
Spool file class	CLASS <i>cl</i>	Device services spool files of the specified classes, in the specified order. An output device can service as many as 8 output classes.

The options for the second line of the response can be:

Category	Possible Values	Meaning
Form	FORM <i>form</i>	Device services the specified 1- to 8-character operator form number.
Scheduling	MANUAL	Operator schedules forms output. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is punched, the punch stops.
	AUTO	Operator wants CP to schedule forms output. The form that is currently active is processed first. After punching all files with the current form, CP selects a new form for the device.
File separator	SEP	File separator prints between output files.
	NOSEP	File separator does not print between output files.
Length limit	LIMIT NONE	CP selects files for punching regardless of the number of records.
	LIMIT <i>nnnnnnnnnn</i>	CP only selects spool files for punching that are less than or equal to the specified number of records. The spool file record number is the number of copies multiplied by the size of the file.

The options for the third line of the response can be:

Category	Possible Values	Meaning
Destination	DEST	Printer services the specified destination values. The printer can service up to 4 destination values. The first value for the device is always OFF if it was not specifically set by either a START command, or at initialization with the RDEVICE statement in the system configuration file. If it was set, the specified destination is shown. If the remaining destination values were not specified, the values are null and are shown as – NONE –. If they were specified, the specified value is shown.

The last line of this response only appears if there is a file active on the card punch. The line in the example above tells you that the punch at the real device number 1F30 is punching the spool file 0042, owned by the user JAMES. The number of records remaining for output in the file is 12K, where K is the number of records in thousands rounded to the nearest 1000. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million. There are two copies remaining for output. The sequence for the file on the punch is 001.

*Response from Boxed Unit Record Devices:* If a unit record that is not dedicated to a user is unavailable, you see a message like:

```
q 000F
RDR 000F BOXED
Ready;
```

000F is the real device number of the card reader that is unavailable. RDR might be replaced by PRT or PUN.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {command|statement} contains extra option(s) starting with *option*
- HCP003E Invalid option - *option*
- HCP006E Invalid device type - {*rdev*|*vdev*|*ldev*}
- HCP007E Invalid userid - *userid*
- HCP009E An invalid address range was specified.
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP021E A [real] device number was not supplied or it is invalid.
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device {*rdev*|*vdev*|*ldev*} does not exist
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command.
- HCP332E Invalid control until type - *rdev*
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1003E {CP module *name*|*rdev*|*ldev*} {is|was} not found.
- HCP1100E DASD *rdev* is not supported as a system-owned device.
- HCP1150E DASD {*rdev*|*vdev*} is not a valid base exposure.
- HCP2259I Device *rdev* is malfunctioning and producing repeated interrupts. The system is attempting to recover.



- HCP2261I Channel Path *chpid*.
- HCP6005E Option *option* is not supported by z/VM.
- HCP6525E External Security Manager is unavailable.

# QUERY RECORDING



## Authorization

Privilege Class: A, B, C, E, F

## Purpose

Use QUERY RECORDING to determine the status of:

- CP data collection for these records:
  - Accounting
  - EREP
- CP data collection for symptom records
- Active retrieval of these records.

## Operands

### REcording

requests the status of data collection for ACCOUNTING, SYMPTOM, CONFIG, and EREP records.

## Usage Notes

1. User IDs displayed in the output have previously connected successfully to the CP system service which supports the type of record being collected (\*ACCOUNT must be specified for ACCOUNT records, \*LOGREC for EREP records, and \*SYMPTOM for SYMPTOM records). The association to the recording function will remain until a cold start of the system is performed.
2. If the PENDING indication is ON and the record count has not decreased, an error has probably occurred, and data retrieval is not proceeding normally. The virtual machine retrieving the records should enter a SYSTEM RESET command and reinitiate data retrieval.
3. Class A and B users receive the status of all recording functions. Other users receive the status of the recording functions for which they are eligible to retrieve data records.

## Responses

Response 1:

RECORDING	COUNT	LMT	USERID	COMMUNICATION
function {ON } {OFF}	nnnnnnnn	mmm	userid	{INACTIVE } {ACTIVE [PENDING]}

### RECORDING

identifies the name of the recording function. ACCOUNT is the name used for accounting records, SYMPTOM is the name used for symptom records, CONFIG is the name used for configuration records, and EREP is used for error recording (for example, LOGREC records).

### ON

### OFF

identifies whether records are being saved for or sent to the userid specified.

**COUNT**

identifies the number of records pending in storage. If the count exceeds 99999999, this will be indicated by 99999999+ in the output.

**LMT**

identifies the threshold limit used to calculate when the system operator and retrieving virtual machine are notified that uncollected records are accumulating.

**USERID**

identifies the virtual machine eligible to retrieve the associated records. This user can alter selected fields in the CP recording table by using the RECORDING command.

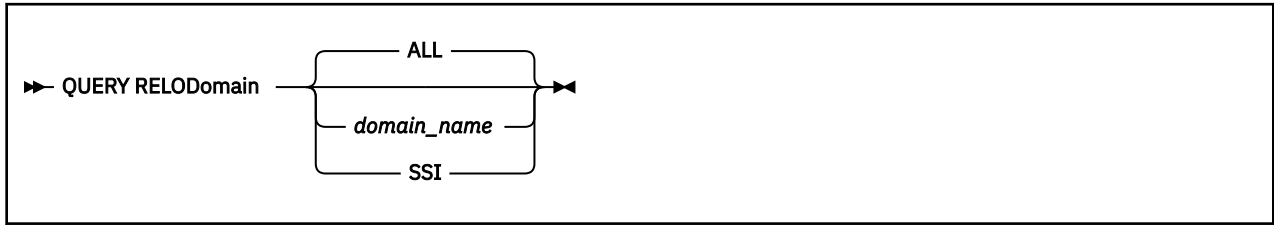
**COMMUNICATION**

is the status of the virtual machine's communication with CP for record retrieval. INACTIVE means that the user must start communication with CP to begin record retrieval. ACTIVE means that the user has already initiated the communication. PENDING means that CP-to-virtual machine communication was initiated and CP sent a record to the virtual machine for retrieval.

**Messages**

- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*

## QUERY RELODOMAIN



### Authorization

Privilege Class: A, B, C, E

### Purpose

Use QUERY RELODOMAIN to list the members of one or more relocation domains. The command will display the SSI members to which a guest with the specified relocation domain may be relocated or will display all of the possible relocation domains and the members of each.

### Operands

#### ALL

specifies that the membership of all domains defined within the SSI cluster be listed. The list will include those implicitly defined domains such as the SSI domain and the domains that contain each single member (singleton domains).

#### SSI

specifies that membership of the SSI relocation domain is to be displayed. The SSI domain contains all members of the SSI cluster.

#### *domain\_name*

is the 1- to 8-character name of the relocation domain whose members are to be displayed.

### Usage Notes

1. The DEFINE RELODOMAIN command can be used to create installation-defined relocation domains.
2. For information about [Using Relocation Domains](#), see *z/VM: CP Planning and Administration*.

### Responses

Responses issued by QUERY RELODOMAIN

```
Domain domain_name members: mem1 mem2 mem3 mem4
This system is not a member of an SSI cluster.
```

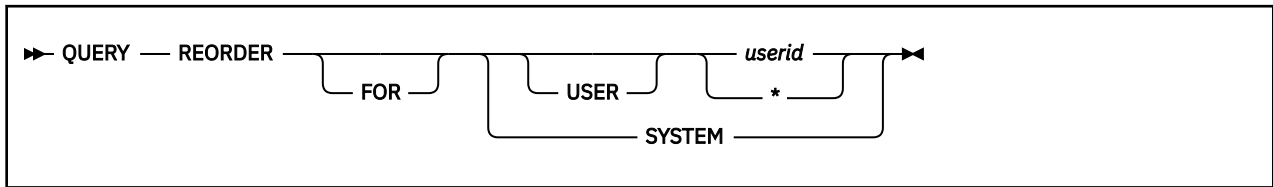
This response can be issued with no members listed if the domain is a singleton domain for a member that has been deleted by a SET SSI SLOT *n* AVAILABLE command. No new virtual machines can be assigned to this domain.

### Messages

- HCP0002E Invalid operand - *operand*
- HCP0003E Invalid option - *option*
- HCP0013E Conflicting option - *option*
- HCP0026E Operand missing or invalid
- HCP1821E Relocation domain *domain\_name* does not exist

- HCP6706E Invalid string - *string*

## QUERY REORDER



### Authorization

Privilege Class: B, E

### Purpose

With the introduction of z/VM memory scalability up to 1T of real memory, the reorder process is no longer supported. The QUERY REORDER command remains for compatibility purposes but issuance of the command always indicates reorder is off.

### Operands

#### FOR USER

are optional keywords to aid in readability of the command.

#### *userid*

displays the reorder setting for the specified virtual machine.

#### \*

displays the reorder setting for the command issuer's virtual machine.

#### SYSTEM

displays the default system-wide setting.

### Usage Notes

1. The SYSTEM setting applies to the Shared Frame Owned list and the User Frame Owned list of every virtual machine in the system with the exception of those virtual machines for which a SET REORDER command was issued to override the SYSTEM default. Issue the QUERY REORDER *userid* command to see the setting for a specific virtual machine.

### Responses

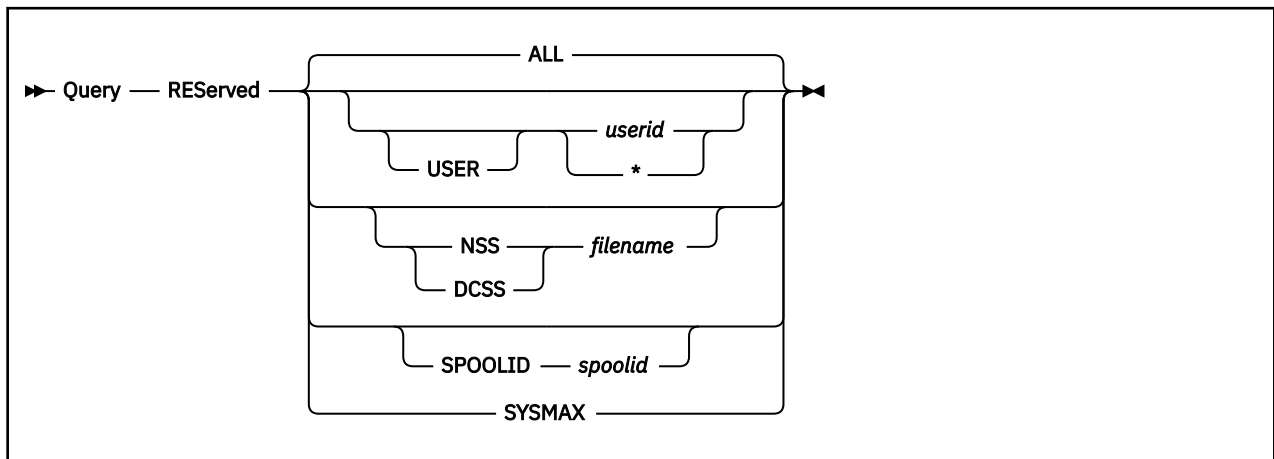
Reorder is ON|OFF for user *userid*

Reorder default is ON|OFF for SYSTEM

### Messages

- HCP002E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E User not logged on

## QUERY RESERVED



### Authorization

Privilege Class: A, E

### Purpose

Use QUERY RESERVED to display the amount of reserved storage, along with the current resident storage and instantiated storage for a virtual machine, Named Saved Systems (NSS), or Discontiguous Saved Segment (DCSS) having a reserved storage setting.

### Operands

#### ALL

displays the reserved settings for all virtual machines, NSSs, and DCSSs, as well as the total for the system. This is the default.

#### USER

an optional keyword you can use when specifying a userid.

#### *userid*

\*

identifies the virtual machine whose reserved storage is to be displayed. To specify your own virtual machine, you may use an asterisk.

#### **NSS filename**

identifies a system data file containing an available NSS.

#### **DCSS filename**

identifies a system data file containing an available DCSS.

#### **SPOOLID spoolid**

identifies a system data file containing an available or pending purge NSS or DCSS.

#### **SYSMAX**

indicates the reserved storage system maximum should be displayed.

### Usage Notes

1. To avoid confusion with the CMS QUERY RESERVED command, you should enter the QUERY RESERVED command with the CP prefix (for example, cp query reserved all) or you should be in CP Read mode when you enter the command.

## QUERY RESERVED

- For readability, the storage values within a single line of the command response are displayed with the same storage unit. The values for the resident and instantiated storage may be rounded to accommodate the unit specification of the reserved setting. The reserved setting will never be rounded although it may be shown in a smaller unit than entered, when the resident or instantiated storage size is less than the reserved setting, in order to maintain the same storage unit across the line.

If you want precise values for the resident and instantiated counts, you should use the INDICATE USER or INDICATE NSS commands.

## Responses

Response 1:

The response to QUERY RESERVED ALL shows all existing reserved storage settings, along with the amount of storage resident and instantiated for those users, NSSs, and DCSSs having a reserved setting.

Resource	Type	SPID	Reserved	Resident	Instantiated
<i>userid1</i>	USER	----	<i>rsvstorU</i>	<i>resstorU</i>	<i>instanstorU</i>
<i>userid2</i>	USER	----	<i>rsvstorU</i>	<i>resstorU</i>	<i>instanstorU</i>
...					
<i>filename1</i>	NSS	<i>spid</i>	<i>rsvstorU</i>	<i>resstorU</i>	<i>instanstorU</i>
<i>filename2</i>	DCSS	<i>spid</i>	<i>rsvstorU</i>	<i>resstorU</i>	<i>instanstorU</i>
...					
	Totals		<i>rsvstorU</i>	<i>resstorU</i>	<i>instanstorU</i>

where:

### ***useridn***

identifies the virtual machine.

### ***filenamem***

identifies the Named Saved System or Discontiguous Saved Segment by name.

### **USER**

### **NSS**

### **DCSS|DCSSG**

identifies the type of resource as a virtual machine, Named Saved System, or Discontiguous Saved Segment.

----

four dashes are displayed to indicate a spool id is not applicable when Type=USER.

### ***rsvstorU***

is the reserved storage setting for this userid in storage units (K,M,G,T,P,E).

If the reserved value was specified as a count of frames on the SET RESERVED command, the value may be rounded to fit within 7-decimal digits plus a storage unit suffix.

### ***resstorU***

is the current resident storage size for this user, NSS, or DCSS in storage units (K,M,G,T,P,E) at the time the command is issued. This value may be rounded.

### ***instanstorU***

is the current instantiated storage size for this user, NSS, or DCSS in storage units (K,M,G,T,P,E) at the time the command is issued. This value may be rounded. Since guest pages can exist in multiple levels of the storage hierarchy (for example, in real storage and in auxiliary storage), the instantiated storage may be less than the storage size calculated by summing the page counts for real and auxiliary storage as shown in response to the INDICATE USER or INDICATE NSS command.

Response 2:

The response to QUERY RESERVED USER shows the reserved setting along with the amount of storage resident and instantiated for a single user.

Resource	Type	SPID	Reserved	Resident	Instantiated
<i>userid</i>	USER	----	<i>rsvstorU</i>	<i>resstorU</i>	<i>instanstorU</i>



Response 3:

The response to QUERY RESERVED NSS | DCSS | SPOOLID shows the reserved setting along with the amount of storage resident and instantiated for an NSS or DCSS.

Resource	Type	SPID	Reserved	Resident	Instantiated
<i>filename</i>	<i>type</i>	<i>spid</i>	<i>rsustorU</i>	<i>resstorU</i>	<i>instanstorU</i>

Response 4:

The response to QUERY RESERVED SYSMAX shows the maximum allowed reserved storage size along with the system maximum as a percentage of DPA and real online storage:

```
SYSMAX = nU    xxx% of pageable storage    yyy% of online storage
```

where:

**nU**

identifies the total maximum allowable amount of reserved storage for all users, NSSs, and DCSSs in storage units (K,M,G,T,P,E).

**xxx%**

specifies the system maximum as an approximate percentage of the Dynamic Paging Area (DPA). The percent value is rounded to the nearest whole percent. When the value is less than .5%, it is shown as <1%.

The size of the DPA varies over time. It includes available frames and those that are pageable at the time the command is issued. A system maximum near the size of DPA can have undesirable effects on performance.

**yyy%**

specifies the system maximum as an approximate percentage of real online storage. The percent value is rounded to the nearest whole percent. When the value is less than .5%, it is shown as <1%.

Response 5:

If a user, NSS, or DCSS does not have a reserved setting, the following is displayed:

```
RESERVED is OFF for userid|filename|SYSMAX
```

or

```
RESERVED is OFF for spoolid spoolid
```

Response 6:

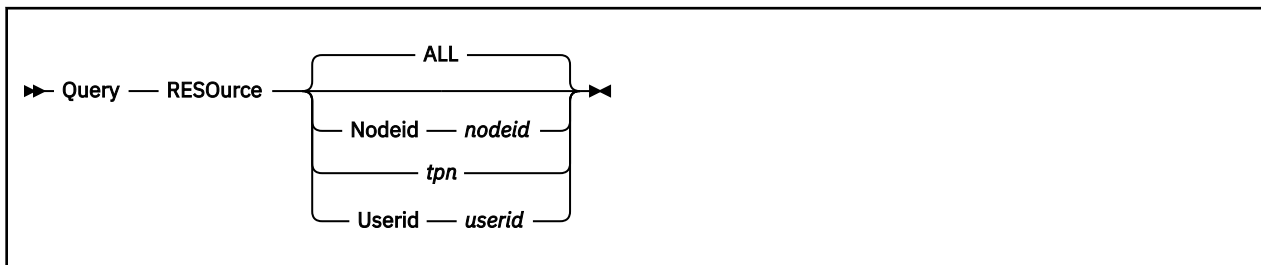
If there are no reserved frames, the following is displayed in response to the QUERY RESERVED ALL command:

```
There are no reserved frames.
```

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP045E *userid* not logged on
- HCP2549E {NSS | DCSS} with {*spoolid spoolid* | *filename filename*} could not be found.
- HCP6704E Missing token at end of line.

## QUERY RESOURCE



### Authorization

Privilege Class: G

### Purpose

Use QUERY RESOURCE to display information about:

- Local, global, and system resources defined on the local z/VM system
- Global resources defined at the other domains in the Communication Services (CS) collection.

### Operands

#### ALL

displays information for all local, global, system resources defined on the local system and the global resources defined in the CS collection.

#### Nodeid *nodeid*

displays information for the global resources that are located at the node *nodeid*.

#### *tpn*

displays information for the specified global resource (transaction program).

#### Userid *userid*

displays information about resources owned by *userid* that are located on the local VM node.

### Responses

Response 1:

```
{Resource: resid} Type: Global Owing Nodeid: nodeid [via tnode]
{Service: stpn }
```

#### *resid*

specifies the resource ID of the global resource known in the CS collection.

#### *stpn*

specifies the hexadecimal name of a service transaction program.

#### *nodeid*

specifies the node ID of the system in the CS collection that owns the resource.

#### *tnode*

is the node through which the resource is known.

Response 2:

```
{Resource: resid} Type: restype Owing Userid: userid
{Service: stpn }
```

**resid**

specifies the resource ID of the local, global, or system resource defined at the local z/VM system.

**stpn**

specifies the hexadecimal name of a service transaction program.

**restype**

indicates if this is a local, global, or system resource.

**userid**

is the user ID on the local system that owns the resource.

Response 3:

```
Status: status.
```

**status**

indicates if an identify or a revoke is pending for the resource. This response is suppressed if the resource is identified.

Response 4:

```
No resources identified.
```

Response 5:

```
No resources identified by node nodeid.
```

**nodeid**

specifies the node ID that has no resources identified.

Response 6:

```
No resources identified by user userid.
```

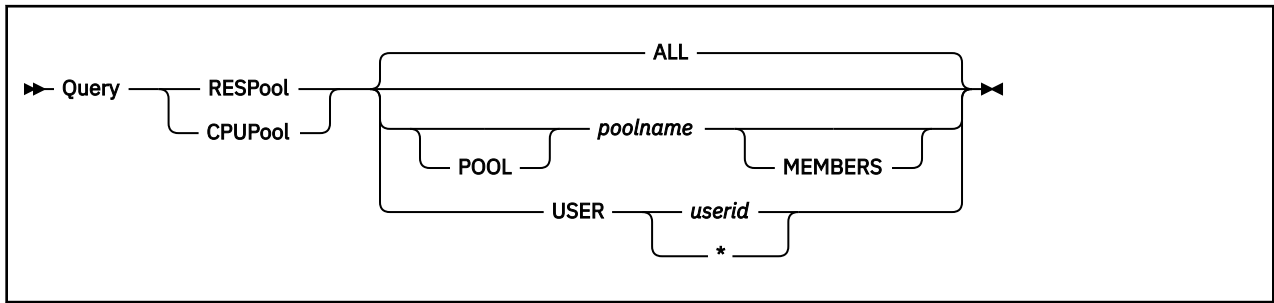
**userid**

is the user ID that has no resources identified.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP2724E Resource *tpn* is undefined.

## QUERY RESPOOL



### Authorization

Privilege Class: A, E

### Purpose

Use QUERY RESPOOL to display information about resource pools that are defined on the system. For compatibility with prior releases, CPUPOOL is accepted as a synonym for RESPOOL.

### Operands

#### ALL

displays information about all resource pools defined on the system. This is the default.

#### *poolname*

#### **POOL** *poolname*

displays the definition of the specified resource pool, which includes the name of the pool, the pool resource limits, the type of CPU resource that is limited, and the number of users assigned to the pool.

#### **MEMBERS**

displays the names of the users assigned to the resource pool.

#### **USER** *userid*

#### **USER** \*

displays the name of the resource pool to which the specified user is assigned. If an asterisk (\*) is specified, the name of the resource pool for the user issuing the command is displayed.

### Usage Notes

1. Because the scope of a resource pool is a single z/VM system, the QUERY RESPOOL command returns information only about the system on which the command was issued. To obtain resource pool information for another z/VM system in an SSI cluster, use the AT command. For example, to display information about all of the resource pools on member VMSYS2 in the SSI cluster, issue the command:

```
at vmsys2 cmd query respool
```

2. If CPU affinity is suppressed for IFL processors and a resource pool in the response has a TYPE of IFL, the QUERY RESPOOL command response will include the following last line:

```
CPU Affinity is suppressed for CPU type IFL
```

3. Because the scope of a resource pool is a single z/VM system, guest relocation essentially removes the user from the resource pool on the source system and assigns the user to the resource pool on the destination system. From the time a VMRELOCATE MOVE command starts until the user is moved to the destination system and can consume its resources, the user is reported on the destination system as an "incoming user" in QUERY responses. The count of resource pool members includes

these incoming users, so the resource pool cannot be deleted. However, the names of incoming users are not displayed in the response to a QUERY RESPOOL *poolname* MEMBERS command because they are not yet consuming or being limited by pool resources. After an incoming user can consume pool resources, its name is included in the QUERY response. There is a short period of time when that user ID will be a skeleton LOGNMxxx ID.

## Responses

Response 1:

The following response is displayed for QUERY RESPOOL ALL if no resource pools are defined:

```
No resource pools exist
```

Response 2:

The following example shows the format of the response to QUERY RESPOOL ALL when resource pools are defined. Each line under the heading contains the definition of a resource pool, which includes the name of the pool, the pool limits, the type of CPU resource that is limited, and the number of users assigned to the pool.

Pool name	CPU	Type	Storage	Trim	Members
LINUXP2	8.00 Cores	IFL	NoLimit	----	0
LINUXP1	NoLimit	IFL	NoLimit	----	6
CPPOOL10	12 %	CP	NoLimit	----	8
LINUXP3	30 %	IFL	NoLimit	----	20

In this example there are four resource pools. Resource pool LINUXP2 has a CAPACITY limit for IFL processors and no storage limit. Resource pool LINUXP1 has no CPU limit and no storage limit. Resource pool CPPOOL10 has a LIMITHARD limit for CP processors and no storage limit. Resource pool LINUXP3 has a LIMITHARD limit for IFL processors and no storage limit.

If a defined resource pool is of type IFL and there are no IFL processors on the system, the following is displayed as the last line of the response:

```
CPU Affinity is suppressed for CPU type IFL
```

Response 3:

The response to QUERY RESPOOL *poolname* displays the definition of the specified resource pool, as in this example:

Pool name	CPU	Type	Storage	Trim	Members
LINUXP1	NoLimit	IFL	NoLimit	----	6

Response 4:

The response to QUERY RESPOOL *poolname* MEMBERS displays the definition of the specified resource pool and lists the members of the pool, five per line, as in this example:

Pool name	CPU	Type	Storage	Trim	Members
LINUXP1	NoLimit	IFL	NoLimit	----	6
The following users are members of resource pool LINUXP1:					
D70LIN12	D79LIN03	D79ADM	D79LIN10	D79LIN07	
D79LIN04					

If there are no IFL processors on the system, the following is displayed as the last line of the response:

```
CPU Affinity is suppressed for CPU type IFL
```

Response 5:

## QUERY RESPOOL

The following example shows the response to `QUERY RESPOOL poolname MEMBERS` if the specified resource pool has no members:

```
Pool name      CPU    Type  Storage Trim Members
LINUXP2      8.00 Cores  IFL  NoLimit ----      0
Resource pool LINUXP2 has no members
```

Response 6:

The following example shows the response to `QUERY RESPOOL poolname MEMBERS` if the specified resource pool has two members that are being relocated from another system. One member has a relocation skeleton user ID (LOGNM012) and can consume pool resources, whereas the other is still incoming.

```
Pool name      CPU    Type  Storage Trim Members
LINUXP4      8.00 Cores  IFL  NoLimit ----      2
Resource pool LINUXP4 has incoming members
The following users are members of resource pool LINUXP4:
LOGNM012
```

Response 7:

The following response to `QUERY RESPOOL USER userid` is displayed if the specified user is assigned to a resource pool:

```
User userid is in resource pool poolname
```

Response 8:

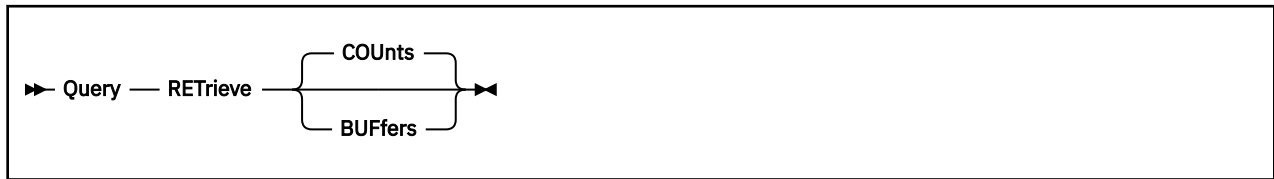
The following response to `QUERY RESPOOL USER userid` is displayed if the specified user is not assigned to any resource pool:

```
User userid is not in a resource pool
```

## Messages

- HCP002E Invalid operand — *operand*
- HCP003E Invalid option — command contains extra option(s) starting with *option*
- HCP007E Invalid userid — *userid*
- HCP045E *userid* not logged on
- HCP1001E An operand is missing for USER
- HCP1001E An operand is missing for POOL
- HCP3151E Resource pool *poolname* does not exist

## QUERY RETRIEVE



### Authorization

Privilege Class: G

### Purpose

Use QUERY RETRIEVE to display the number of program function (PF) key RETRIEVE buffers available to your virtual machine and the maximum number of program function key RETRIEVE buffers available to users on the system. You can also use QUERY RETRIEVE to display the contents of the program function key RETRIEVE buffers.

### Operands

#### COUnts

displays the number of program function (PF) key RETRIEVE buffers available to your virtual machine and the maximum number of program function key RETRIEVE buffers available to users on the system. COUNTS is the default.

#### BUffers

displays the contents of the program function key RETRIEVE buffers.

### Usage Notes

1. CP initially sets the number of buffers available to your virtual machine to the system default, which is defined in the system configuration file with the FEATURES statement. If a default has not been specified in the system configuration file, CP allocates seven RETRIEVE buffers for your virtual machine. For more information on the system configuration file, see the [FEATURES Statement](#) in *z/VM: CP Planning and Administration*.
2. Use the SET RETRIEVE command to request a certain number of PF key RETRIEVE buffers. If you are a class C or class E user, you can use the SET RETRIEVE command to change the maximum number of PF key RETRIEVE buffers available to all users.

For more information, see [“SET RETRIEVE”](#) on page 1681.

3. It is possible that the number of PF key RETRIEVE buffers allocated for your virtual machine may exceed the maximum number established for other users on the system. When the maximum number of PF key RETRIEVE buffers available to users is decreased, buffers already allocated to users are not deleted.

### Responses

Response 1:

If you enter QUERY RETRIEVE COUNTS, you receive the following response:

```
nnn buffers available. Maximum of mmm buffers may be selected.
```

The number of buffers available to your virtual machine will be zero if you have not set a PF key for the retrieve function or defined a value with the SET RETRIEVE command.

## QUERY RETRIEVE

Response 2:

If you enter QUERY RETRIEVE BUFFERS, you receive the following response:

Position	Contents....
7	<i>cmd7</i>
6	<i>cmd6</i>
5	<i>cmd5</i>
4	<i>cmd4</i>
3	<i>cmd3</i>
2	<i>cmd2</i>
1	<i>cmd1</i>

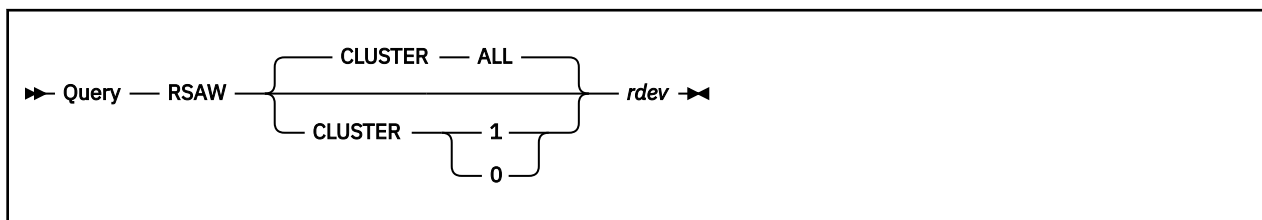
The most recent command is displayed last.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*



## QUERY RSAW



### Authorization

Privilege Class: B

### Purpose

Use QUERY RSAW to generate and display the remote service access word (RSAW) for storage clusters in the DASD subsystem.

### Operands

**CLUSTER ALL**

**CLUSTER 1**

**CLUSTER 0**

indicates the storage clusters from which the remote service access word is to be fetched. ALL is the default.

**rdev**

is the real device number of one of the devices in the subsystem.

### Usage Notes

1. The Storage Subsystem Support Facility supports subsystem error recovery, maintenance, diskette drive operations, and the operator panel. The remote service access word is generated by the Support Facility and is recognized by that Support Facility to allow remote maintenance attachment.
2. This command may be issued to offline devices.

### Responses

Response 1:

```
rdev RSAW is nnnn for cluster n, serial number sssss.
```

This response is repeated for each storage cluster if CLUSTER ALL is specified.

**rdev**

is the real device number specified in the command.

**nnnn**

is the requested remote service access word.

**n**

is the storage cluster number. Storage clusters contain storage directors and storage paths.

**sssss**

is the storage control unit serial number.

Response 2:

```
rdev Subsystem support facility is unavailable.
```

This response is repeated for each storage cluster if CLUSTER ALL is specified.

***rdev***

is the real device number specified in the command.

Response 3:

```
rdev Remote service is not enabled.
```

This response is repeated for each storage cluster if CLUSTER ALL is specified.

***rdev***

is the real device number specified in the command.

### Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*
- HCP1109E The *command* command cannot be processed for device *rdev* because the device is the secondary in a duplex pair.

## QUERY SCMBKS

► Query — SCMBKS ◄

### Authorization

Privilege Class: B, E

### Purpose

Use QUERY SCMBKS to display the:

- Total number of subchannel measurement blocks (SCMBKs) that CP allocated, and
- Number (and percentage) of SCMBKs that are not currently being used.

### Usage Notes

**Format-1 measurement blocks:** When format-1 measurement blocks are being used, measurement blocks are allocated dynamically.

1. SCMBKs are used to accumulate data about a subchannel, such as rate and utilization information. This subchannel measurement data, which CP Monitor collects in various monitor records, can be useful for analyzing the I/O performance characteristics of a real device.

**Format-0 measurement blocks:** See the notes below.

1. During initialization, CP allocates 1 SCMBK for every subchannel on the system and rounds the amount of storage up to the nearest page boundary. Thus, you have enough SCMBK space for every real device on your system plus some free SCMBK space. How much free SCMBK space you have will vary, depending upon how many real devices you have on your system.
2. You should plan for additional SCMBK space if you intend to dynamically add new real devices to your running system. At initialization, you can allocate more free SCMBK space than CP would normally allocate using the SCMBK operand of the STORAGE statement in the system configuration file. For more information, see [STORAGE Statement](#) in *z/VM: CP Planning and Administration*.
3. If you run out of free SCMBK space on a running system and you need to measure additional real devices, you have two choices:
  - a. Re-IPL and specify more free SCMBK space using the SCMBK operand, or
  - b. Stop collecting subchannel measurement data for one or more devices. When you use one of the following commands:
    - DELETE DEVICE (see [“DELETE DEVICE / IODEVICE”](#) on page 265)
    - DELETE IODEVICE (see [“DELETE DEVICE / IODEVICE”](#) on page 265)
    - DELETE RDEVICE (see [“DELETE RDEVICE”](#) on page 274)
    - SET SCMEASURE OFF (see [“SET SCMEASURE”](#) on page 1685)

CP returns the SCMBK assigned to that device (or devices) to the pool of available SCMBKs. It is then available to be used to collect subchannel measurement data for another device.

### Responses

To see information about the number of SCMBKs that CP allocated and information about the free SCMBK space, enter the following QUERY SCMBKS command:

## QUERY SCMBKS

Response 1:

```
query scmbks
Total SCMBKs= 1024 (pages= 8), Free SCMBKs= 77 (7%)
Ready;
```

This response tells you that CP allocated 1024 SCMBKs (which takes up 8 pages of storage) and all but 77 of the SCMBKs are in use, which is roughly 7% of the total SCMBK space.

Response 2:

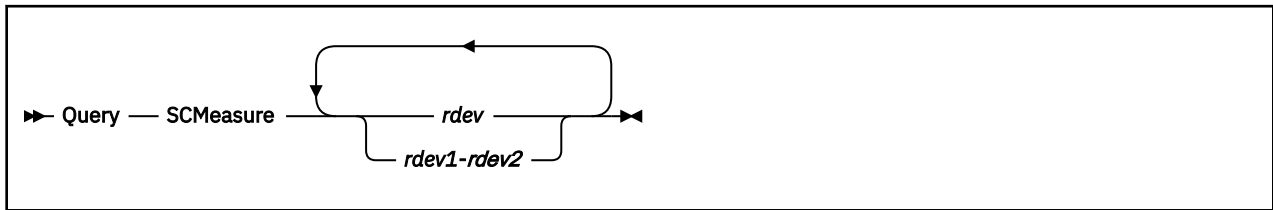
```
query scmbks
Total SCMBKs= 31 (pages= 1), Free SCMBKs= 0 (0%) dynamically allocated
Ready;
```

This response tells you that CP allocated 31 SCMBKs (which took up 1 page of storage). Dynamically allocated means format-1 measurement blocks are being used. In this case, free SCMBKs will always be zero.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with - *option*

## QUERY SCMEASURE



### Authorization

Privilege Class: B, E

### Purpose

Use QUERY SCMEASURE to display whether subchannel measurement data is being collected for a specific device or range of devices.

### Operands

**rdev**

**rdev1-rdev2**

is the real device number (or numbers) of the device for which you want to display subchannel measuring status. The variable *rdev* must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single device address, a range of device addresses, or any combination thereof.

### Usage Notes

1. Subchannel measurement blocks (SCMBKS) are used to accumulate data about a subchannel, such as rate and utilization information. This subchannel measurement data, which CP Monitor collects in various monitor records, can be useful for analyzing the I/O performance characteristics of a real device.
2. To enable or disable the collecting of subchannel measurement data for one or more devices, use the SET SCMEASURE command (see [“SET SCMEASURE” on page 1685](#)).
3. When a device is first initialized, the default is to enable it for collecting subchannel measurements. To change this initial default, use the SCMEASURED or NOTSCMEASURED operands of the DEVICES statement in the system configuration file. For more information about the [DEVICES Statement](#), see *z/VM: CP Planning and Administration*.
4. Error messages produced for this command may vary depending on how the device was specified on the command line.

When a single device is listed on the command line, an error will be reported if there is no query information available for the single device. For example:

```
query scmeasure 102
Device 0102 does not exist
Ready(00040);
```

When a range of devices is listed on the command line, only devices that have query information available are shown. Error messages are not issued for individual devices within the range. For example:

```
query scmeasure 100-105
Subchannel measuring for device 103 is on
Subchannel measuring for device 104 is off
Ready;
```

## QUERY SCMEASURE

If query information is not available for at least one device in a range, then a generic error message will be issued. For example:

```
query scmeasure 100-102
HCPCHC1003E 0100-0102 was not found
Ready (01003);
```

## Responses

Response 1:

To find out whether CP is collecting subchannel measurement data for device 131, enter the following QUERY SCMEASURE command:

```
query scmeasure 131
Subchannel measuring for device 0131 is on
Ready;
```

Response 2:

To find out which devices CP is collecting subchannel measurement data for in the device range 750 through 759, enter the following QUERY SCMEASURE command:

```
query scmeasure 750-759
Subchannel measuring for device 0750 is off
Subchannel measuring for device 0751 is off
Subchannel measuring for device 0752 is on
Subchannel measuring for device 0753 is on
Subchannel measuring for device 0754 is on
Subchannel measuring for device 0755 is off
Subchannel measuring for device 0756 is on
Subchannel measuring for device 0757 is on
Ready;
```

This example shows that:

- CP is not collecting subchannel measurement data for devices 750, 751, and 755
- CP is collecting subchannel measurement data for devices 752, 753, 754, 756, and 757

## Messages

- HCP003E Invalid option - *option*
- HCP009E Invalid range - *rdev-rdev*
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device *rdev* does not exist
- HCP1003E *rdev* was not found.
- HCP6706E Invalid RDEV - *rdev*. RDEV not initialized

## QUERY SCREEN

► Query — SCReen ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY SCREEN to display the extended color and highlight values governing your console.

### Usage Notes

1. The extended color and highlighting values are set either by the SCREEN statement in your directory entry or when you enter the SCREEN command.
2. If you do not have a SCREEN directory statement in your directory entry and you have not issued the SCREEN command during the current session, the response to the QUERY SCREEN command is DEFAULT for the color value and NONE for the extended highlight value.

### Responses

Response 1:

CPOUT	<i>color</i>	<i>exthilight</i>	VMOUT	<i>color</i>	<i>exthilight</i>
INREDISP	<i>color</i>	<i>exthilight</i>	INAREA	<i>color</i>	<i>exthilight</i>
STATAREA	<i>color</i>	<i>exthilight</i>			

#### ***color***

identifies the color value currently in effect for your virtual console.

#### ***exthilight***

identifies the extended highlight value currently in effect for your virtual console.

### Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*

## QUERY SDF

► Query — SDF ◄

### Authorization

Privilege Class: A, B, C, D, E, G

### Purpose

Use QUERY SDF to display a count of all system data files.

### Usage Notes

1. Counts of the various types of system data files are displayed only to specific classes of users:
  - a. IMG files to class A, B, C, D, and E users
  - b. NSS files to class E users
  - c. UCR files to class A, B, and C users
  - d. NLS files to class E users
  - e. TRF files to class A, C, D, E, and G users
  - f. For a class G user, only the number of TRF files that the class G user owns is displayed.
2. QUERY SDF ALL returns the same information as QUERY SDF.

### Responses

Response 1:

```
FILES: [ {nnnn} IMG ] [ , {nnnn} NSS ] [ , {nnnn} UCR ] [ , {nnnn} TRF ] [ , {nnnn} NLS ]
       [ { NO} ] [ { NO} ] [ { NO} ] [ { NO} ] [ { NO} ]
```

#### **nnnn**

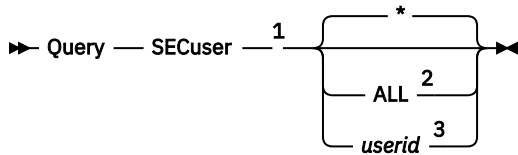
is the number of system data files of each type.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP849E A *userid* must be specified with the spoolid
- HCP1558E Command processing cannot complete.



## QUERY SECUSER



### Notes:

- <sup>1</sup> If QUERY SECUSER is executed with the AT command, the ALL or *userid* operand is required.
- <sup>2</sup> ALL is not valid for class G users.
- <sup>3</sup> A class G user may specify only the user's own user ID.

## Authorization

Privilege Class: A, B, C, G

## Purpose

Use QUERY SECUSER to determine the secondary user setting for a virtual machine (the primary user). CP displays the identity and status of the secondary user. If you are a class G user, you can display the secondary user setting for your own virtual machine. If you are a class A, B, or C user, you can display the secondary user setting for any virtual machine or for all local virtual machines that have secondary users defined.

## Operands

**\***

displays the secondary user setting for the user who issues the command. This is the default.

**ALL**

displays the secondary user setting for all local users currently logged on or disconnected that have secondary users defined.

***userid***

identifies the user whose secondary user setting is to be displayed.

## Usage Notes

1. The initial setting for your secondary user ID is determined by the CONSOLE statement in your user directory entry.
2. A virtual machine cannot have both a secondary user and an observer defined by any combination of the SET SECUSER and SET OBSERVER commands and the CONSOLE directory statement.
3. In an SSI cluster, the primary user and secondary user can be logged on to different member systems.

If either user is a multiconfiguration virtual machine:

- The secondary user will function in that capacity only when it is local (logged on to the same member as the primary user). If the primary user is a multiconfiguration virtual machine, the secondary user must be logged on to the same member as the instance of the primary user for which the secondary user setting was established:
  - If the setting was established by a CONSOLE statement in the identity entry in the primary user's virtual machine definition, a potential secondary user relationship exists for every member instance of the primary user.

## QUERY SECUSER

- If the setting was established by a CONSOLE statement in a subconfiguration entry in the primary user's virtual machine definition, a potential secondary user relationship exists for that member instance of the primary user.
  - If the setting was established by a SET SECUSER command, a potential secondary user relationship exists for the member on which the SET SECUSER command was issued.
  - If the secondary user is remote:
    - The secondary user can issue SEND commands to the primary user. The AT *sysname* operands are required if the primary user is a multiconfiguration virtual machine instance.
    - The secondary user will not receive responses to SEND commands or any other output from the primary user.
4. If your virtual machine requests console input while disconnected and you do not have a functional secondary user, CP waits for some installation-defined period of time (usually 15 minutes) for you to reconnect. If you do not reconnect within that time, CP logs you off the system.

### Responses

Responses 1 through 5 are returned if you issue the QUERY SECUSER command without any additional operands.

Response 1:

```
SECONDARY USER userid IS DISCONNECTED
```

is returned if a secondary user is defined for your virtual machine (on the CONSOLE statement in your virtual machine definition or on a SET SECUSER command), and that user is disconnected.

If you disconnect from the system, the result is the same as if no secondary user ID was defined, unless the secondary user's virtual machine handles console I/O through the message system service. In this case, secondary console support is provided to your disconnected virtual machine. If the secondary user reconnects, it can provide secondary console functions for your virtual machine.

Response 2:

```
SECONDARY USER userid IS NOT LOGGED ON
```

is returned if a secondary user is defined for your virtual machine (on the CONSOLE statement in your virtual machine definition or on a SET SECUSER command), and that user is not logged on.

If you disconnect from the system, the result is the same as if no secondary user ID was defined. If the secondary user logs on, it can provide secondary console functions for your virtual machine.

Response 3:

```
SECONDARY USER userid IS LOGGED ON
```

is returned if a secondary user is defined for your virtual machine (on the CONSOLE statement in your virtual machine definition or on a SET SECUSER command), and that user is capable of providing secondary console support for you.

If you, the primary user, disconnect from the system, that secondary user receives your console output until you reconnect or log off. The secondary user may provide console input to your virtual machine or to CP on your behalf by using the SEND command.

Response 4:

```
SECONDARY USER IS UNDEFINED
```

is returned if a secondary user is not defined for your virtual machine. If you disconnect from the system, your console output is lost until you reconnect. If you have spooled your console output, it is retained in your console spool file.

Response 5:

```
SECONDARY USER IS NOT LOGGED ON LOCALLY
```

is returned if all of the following conditions are true:

- This system is a member of an SSI cluster.
- You, the primary user, are defined as a multiconfiguration virtual machine.
- A secondary user is defined for your virtual machine on this member of the cluster (on a CONSOLE statement in the subconfiguration entry for this member in your virtual machine definition, or on a SET SECUSER command issued on this member).
- That user is not logged on to this member of the cluster. (The user might be logged on to another member.)

No console output from your virtual machine will be sent to the secondary user. See usage note [“3” on page 1153](#).

Response 6:

```
FAILED MAC CHECK
```

is returned if security label checking by an external security manager (ESM) is active and the primary/secondary user relationship fails the mandatory access control checks.

Response 7: If you issue the QUERY SECUSER command with an additional option and the secondary user is disconnected, CP responds with a display that looks like this:

```

Userid      Secondary
MAINT      Userid    Status
          DAVID  disconnected
    
```

The Status column in the display might also contain "not logged on", "logged on", "not defined", "not logged on locally", or "failed MAC check".

Response 8: If a class A, B, or C user issues QUERY SECUSER ALL, CP displays the current secondary user settings for all users that are currently logged on or disconnected. If no currently logged on users have a secondary user defined, CP issues the following response:

```
No secondary users defined or active.
```

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP045E *userid* not logged on
- HCP1149E Error communicating with another SSI member. It cannot be determined if *command* executed remotely.

## QUERY SET

► Query — Set ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY SET to display the status of many SET command functions.

### Operands

#### Set

requests a display of the status of the SET command functions.

### Responses

Response 1:

```

MSG {ON } , WNG {ON } , EMSG {ON } , ACNT OFF , RUN {ON }
  {OFF }      {OFF }      {OFF }      {OFF }
  {IUCV}      {IUCV}      {CODE}
                          {TEXT}
                          {IUCV}

LINEDIT {ON } , TIMER {ON } , ISAM OFF , ECMODE ON
  {OFF }      {OFF }
              {REAL}

ASSIST OFF , PAGEX {ON } , AUTOPOLL {ON }
                {OFF }      {OFF }

IMSG {ON } , SMSG {ON } , AFFINITY NONE , NOTRAN OFF
  {OFF }      {OFF }
  {IUCV}      {IUCV}

VMSAVE OFF , 370E OFF

STBYPASS OFF , STMULTI OFF 00/000

MIH {ON } , VMCONIO {OFF } , CPCONIO {OFF } , SVCACCL OFF , CONCEAL {ON }
  {OFF }      {IUCV}      {IUCV}      {OFF }

MACHINE {ESA} , SVC76 {CP} , NOPDATA {ON }
  {XA }      {VM}      {OFF }
  {XC }
  {Z }

CCWTRAN ON , 370ACCOM {ON } , TIMEBOMB {IDLE } , DIALDROP {ON }
  {OFF }      {OFF }      {ENABLED}
  {ACTIVE }

```

#### **370ACCOM ON**

#### **370ACCOM OFF**

indicates whether 370-only CMS applications can run in XA, ESA, or XC virtual machines. Note that a guest in z/Architecture mode or z/XC mode can not have 370ACCOM on.

#### **370E OFF**

appears for compatibility purposes.

**ACNT OFF**

appears for compatibility purposes.

**AFFINITY NONE**

appears for compatibility purposes.

**ASSIST OFF**

appears for compatibility purposes.

**AUTOPOLL ON****AUTOPOLL OFF**

indicates whether CP or your virtual machine detects changes to virtual autopolling channel programs. ON specifies that your virtual machine must notify CP that a change has been made to a virtual autopolling channel program. OFF specifies that CP tests for changes to virtual autopolling channel programs.

**CCWTRAN ON**

appears for compatibility purposes.

**CONCEAL ON****CONCEAL OFF**

indicates whether the virtual machine is running in a protected application environment. ON indicates that certain error conditions and virtual machine state changes detected by CP will result in an automatic re-IPL of the virtual machine. OFF indicates that the virtual machine will stay in whatever environment it was in when the error occurred.

**CPCONIO OFF****CPCONIO IUCV**

controls the destination of any CP-generated output. OFF indicates that the output is displayed at the terminal. IUCV indicates that the output is sent to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, data is handled as if CPCONIO were set to OFF.

**DIALDROP ON****DIALDROP OFF**

indicates whether devices DIAled to your virtual machine are dropped (ON) or stay connected (OFF) when a reset occurs (for example, when an IPL or SYSTEM CLEAR command is issued). Such a device is always dropped when an explicit DETACH or RESET command is issued for the associated virtual device.

**ECMODE ON**

appears for compatibility purposes.

**EMSG ON****EMSG OFF****EMSG CODE****EMSG TEXT****EMSG IUCV**

indicates how error messages are to be edited and displayed. ON indicates that both the error code and the message text are displayed. OFF indicates that no error messages will be displayed. CODE indicates that only the error code is displayed. TEXT indicates that only the error message text are displayed. IUCV indicates that both the error code and error message text are passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the error message is handled as if EMSG were set ON.

**IMSG ON****IMSG OFF****IMSG IUCV**

indicates whether informational messages associated with certain CP commands are to be displayed. ON specifies that informational responses are displayed. OFF specifies that the responses are not displayed. IUCV specifies that the informational responses are passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the responses are handled as if IMSG were set ON.

## QUERY SET

### **ISAM OFF**

appears for compatibility purposes.

### **LINEDIT ON**

### **LINEDIT OFF**

indicates whether line-editing functions and symbols are to be used to edit virtual console input. ON indicates that the system line editing functions are used. OFF indicates that no character or line editing is used for the virtual machine operating system.

### **MACHINE ESA**

### **MACHINE XA**

### **MACHINE XC**

### **MACHINE Z**

indicates the architecture setting of your virtual machine:

#### **ESA**

indicates an ESA virtual machine, which simulates ESA/390 architecture. A guest can switch the virtual machine between ESA/390 and z/Architecture.

#### **XA**

indicates an XA virtual machine, which is functionally equivalent to an ESA virtual machine.

#### **XC**

indicates an XC virtual machine, which simulates ESA/XC architecture. The guest can switch the virtual machine between ESA/XC and z/XC architecture.

#### **Z**

indicates a z/Architecture virtual machine, which simulates only the z/Architecture.

**Note:** An ESA (or XA) virtual machine runs in either full ESA/390 mode or ESA/390 compatibility mode, depending on the level of ESA/390 capability available in the machine where the virtual machine is logged on or in the virtual machine's relocation domain.

### **MIH ON**

### **MIH OFF**

ON specifies that CP handles any detected missing interrupt conditions. OFF specifies that CP does *not* handle any detected missing interrupt conditions.

### **MSG ON**

### **MSG OFF**

### **MSG IUCV**

indicates whether messages sent from other users by means of the CP MESSAGE command are to be displayed. ON specifies that messages are displayed. OFF specifies that messages are not displayed. IUCV specifies that messages are passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the message is handled as if MSG were set ON.

### **NOPDATA ON**

### **NOPDATA OFF**

controls whether NOP CCWs are simulated as data transfer CCWs for spooled punches and printers of your virtual machine. ON indicates that for more than one byte of data, CP attempts to transfer the data addressed by a NOP CCW to the spool file. OFF indicates that CP simulates the NOP the same way a real device and channel does.

### **NOTRAN OFF**

appears for compatibility purposes.

### **PAGEX ON**

### **PAGEX OFF**

indicates whether the pseudo page fault facility is activated for your virtual machine. ON specifies that the facility is activated. OFF specifies the facility is deactivated.

### **RUN ON**

### **RUN OFF**

indicates whether your virtual machine begins to run after you enter something from your terminal, and also indicates whether it runs when a CP READ status is showing. ON indicates that your virtual

machine runs when CP READ status is showing and switches to RUNNING following the entry. OFF specifies that your virtual machine does not run when CP READ status is showing, and will remain in CP READ status following the entry.

**SMSG ON****SMSG OFF****SMSG IUCV**

specifies whether a virtual machine is receiving special messages. OFF indicates the virtual machine cannot receive special messages. ON indicates the virtual machine can receive special messages. IUCV indicates the special messages are passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the special messages are handled as if SMSG were set OFF.

**STBYPASS OFF**

appears for compatibility purposes.

**STMULTI OFF**

appears for compatibility purposes.

**SVC76 CP****SVC76 VM**

indicates how SVC 76 instructions issued by a virtual machine are processed by CP. The designation CP specifies that the virtual machine's error records are recorded by CP error recording facilities, if the virtual machine is running in supervisor state. The designation VM specifies that no errors are recorded and all SVC 76 operations are processed by the virtual machine.

**SVCACCL OFF**

appears for compatibility purposes.

**TIMEBOMB IDLE****TIMEBOMB OFF****TIMEBOMB ENABLED****TIMEBOMB ACTIVE**

displays the dynamic virtual machine time bomb setting. ENABLED means that the time bomb function is initialized but no expiration time is currently set. ACTIVE means that the time bomb function is initialized and an expiration time is currently set. IDLE means that the time bomb function is not initialized. OFF means that the time bomb function is initialized but has been disabled by a SET TIMEBOMB OFF command.

**TIMER ON****TIMER OFF****TIMER REAL**

indicates whether the virtual interval timer is to be updated. ON specifies that the virtual interval timer is updated during virtual processor run time. OFF specifies that the virtual interval timer is not updated. REAL specifies that the virtual interval timer is updated during virtual processor run time and also during virtual wait time. SET 370ACCOM must be set to ON.

**VMCONIO OFF****VMCONIO IUCV**

controls the destination of any virtual machine generated output. This includes any CMS error messages and any line mode I/O done by the virtual machine to the virtual console. OFF indicates that output is displayed at the terminal. IUCV indicates that output is sent to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, data is handled as if VMCONIO were set to OFF.

**VMSAVE OFF**

appears for compatibility purposes.

**WNG ON****WNG OFF****WNG IUCV**

indicates whether warning messages sent from an authorized user through the WARNING command are received at your display. ON specifies that warning messages are displayed. If OFF is specified, no warning messages are displayed. IUCV specifies that warning messages are passed to the virtual

## QUERY SET

machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the warning message is handled as if WNG were set ON.

### Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*



## QUERY SHARE

```
► Query — SHARE — userid ◄
```

### Authorization

Privilege Class: A, E

### Purpose

Use QUERY SHARE to display the type and intensity of a user's share of the system. The share values are displayed for all processor types.

### Operands

#### *userid*

is the virtual machine whose share you want to display.

### Usage Notes

- You can set a user's system-resource-access priority for each processor type with the SET SHARE command, but some of these processor types are not available in certain processor configurations. The share values are still displayed for the unavailable processor types, but they are not used.

### Responses

Response 1:

The following response is returned to the command issuer:

```
USER userid: CP   share_type SHARE = value
                MAXIMUM SHARE = option share_type value
             IFL share_type SHARE = value
                MAXIMUM SHARE = option share_type value
             ICF share_type SHARE = value
                MAXIMUM SHARE = option share_type value
             ZIIP share_type SHARE = value
                MAXIMUM SHARE = option share_type value
```

#### **ABSOLUTE SHARE = *nnn%***

appears for users with an absolute share. The operand *nnn%* specifies the percent of system resources this user is allowed to access.

#### **RELATIVE SHARE = *nnnnn***

appears for users with a relative share. The operand *nnnnn* specifies the relative priority of access to system resources, when compared to all other users with relative shares.

#### ***m* DEDICATED CPU[S]**

specifies the number of virtual CPUs of this type that this user has that have a real processor dedicated to them. This string does not appear if the user has no dedicated processors (for example, when *m* = 0). Because only processors of the primary real processor type can be dedicated, this string can only appear on the response line for virtual CPUs of type CP or IFL.

#### ***option***

either NOLIMIT, LIMITSOFT, or LIMITHARD. Note: When NOLIMIT appears, the type and value will not be shown.

## QUERY SHARE

### ***share\_type***

either RELATIVE or ABSOLUTE.

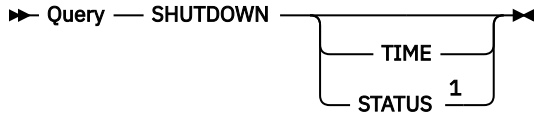
### ***value***

when *type* is RELATIVE, *value* is *nnnn*. When *type* is ABSOLUTE, *value* is *nnn%*.

## **Messages**

- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on

## QUERY SHUTDOWN



Notes:

<sup>1</sup> A user with privilege class G but not privilege class A or C can issue QUERY SHUTDOWN STATUS only.

### Authorization

Privilege Class: A, C, G

### Purpose

Use QUERY SHUTDOWN to learn more about a system shutdown that is in progress. When no operands are specified, the output of the TIME operand followed by the STATUS operand is given. Privilege class A or C is required to issue QUERY SHUTDOWN with no operands.

### Operands

#### TIME

displays the amount of time reserved for a VM SHUTDOWN to be performed. When a SHUTDOWN command is issued or the VM system shuts down automatically as the result of receiving a hardware deactivation signal, there is a time interval within which the shutdown must be completed. The SHUTDOWN TIME value specifies how much of that interval is reserved for the VM Control Program. The difference between these two values determines how much time, if any, guests are allowed to perform their own shutdown processing. Privilege class A or C is required to issue QUERY SHUTDOWN TIME.

#### STATUS

indicates whether a system shutdown is in progress. If so, return information on whether a user ID or a hardware deactivation initiated the system shutdown, the time the shutdown was started, and the amount of time left for guests or servers to shut down. Privilege class A, C, or G is required to issue QUERY SHUTDOWN STATUS.

If the system shutdown was initiated by a hardware deactivation, the message indicates that SYSTEM initiated the shutdown.

### Usage Notes

1. The QUERY SHUTDOWN TIME command displays the same output as the QUERY SHUTDOWNTIME command.
2. If the system shutdown was initiated by a SHUTDOWN command, the output indicates that the shutdown was initiated by "a user" or displays the actual user ID that issued the SHUTDOWN command based on the privilege class of the user ID issuing the QUERY command. Issuers of the QUERY command with privilege class A or C will see the user ID that issued the SHUTDOWN. Issuers without privilege class A or C will see output indicating that "a user" has initiated the shutdown. The user who initiated the system shutdown is the one who has privileges to issue commands that perform those functions.
3. The QUERY SHUTDOWN STATUS command is meant to be used by virtual machines to obtain more information when they receive a shutdown signal.

## QUERY SHUTDOWN

- The output of the QUERY SHUTDOWN STATUS command shows the status of the current system shutdown. It is a point in time and the status can change just after the command is issued. For example, if a SHUTDOWN command issued the shutdown signal, but a SHUTDOWN CANCEL was processed before the QUERY SHUTDOWN STATUS, an issuer of this command could see output indicating that "System shutdown is not in progress".

### Responses

Response 1:

```
SHUTDOWN initiated at yyyy-mm-dd hh:mm:ss by SYSTEM
Signaled users have nnnnn seconds left to shut down
```

is displayed when the QUERY SHUTDOWN STATUS command is accepted and a shutdown initiated by a hardware deactivation is in progress. The privilege class of the issuer does not affect the output. The number of seconds left is in the range 1-65535.

Response 2:

```
System shutdown time: interval seconds
SHUTDOWN initiated at yyyy-mm-dd hh:mm:ss by userid
Signaled users have nnnnn seconds left to shut down
```

is displayed when a QUERY SHUTDOWN command is accepted, a shutdown initiated by the SHUTDOWN command is in progress, and the issuer of the QUERY SHUTDOWN command has privilege class A or C. The *interval* is the amount of time reserved for VM SHUTDOWN processing. The number of seconds left is in the range 1-65535.

Response 3:

```
System shutdown time: interval seconds; previous shutdown duration:
length seconds
SHUTDOWN initiated at yyyy-mm-dd hh:mm:ss by userid
Signaled users have nnnnn seconds left to shut down
```

is displayed when a QUERY SHUTDOWN command is accepted, a shutdown initiated by the SHUTDOWN command is in progress, and the issuer of the QUERY SHUTDOWN command has privilege class A or C. The *interval* is the amount of time reserved for VM SHUTDOWN processing and *length* is the amount of time the previous VM SHUTDOWN required. The number of seconds left is in the range 1-65535.

Response 4:

```
System shutdown time: interval seconds
```

is displayed when a QUERY SHUTDOWN TIME command is accepted. The *interval* is the amount of time reserved for VM SHUTDOWN processing.

Response 5:

```
System shutdown time: interval seconds; previous shutdown duration:
length seconds
```

is displayed when a QUERY SHUTDOWN TIME command is accepted. The *interval* is the amount of time reserved for VM SHUTDOWN processing and *length* is the amount of time the previous VM SHUTDOWN required.

Response 6:

```
SHUTDOWN initiated at yyyy-mm-dd hh:mm:ss by a user
Signaled users have nnnnn seconds left to shut down
```

is displayed when the QUERY SHUTDOWN STATUS command is accepted, a shutdown initiated by the SHUTDOWN command is in progress, and the issuer of the QUERY SHUTDOWN command does not have privilege class A or C but has privilege class G. The number of seconds left is in the range 1-65535.

Response 7:

```
System shutdown is not in progress
```

is displayed when the QUERY SHUTDOWN STATUS command is accepted and no system shutdown is in progress. If the user received a shutdown signal, this response indicates that the user was told to shut down (most likely from a SIGNAL SHUTDOWN or FORCE), but the whole system is not shutting down.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP003E Invalid option - *option*
- HCP026E Operand missing or invalid

## QUERY SHUTDOWNTIME

► Query — SHUTDOWNTime ◄

### Authorization

Privilege Class: A, C

### Purpose

Use QUERY SHUTDOWNTIME to display the amount of time reserved for a VM SHUTDOWN to be performed. When a SHUTDOWN command is issued or the VM system shuts down automatically as the result of receiving a hardware deactivation signal, there is a time interval within which the shutdown must be completed. The SHUTDOWNTIME value specifies how much of that interval is reserved for the VM Control Program, and the difference between these two values determines how much time, if any, guests are allowed to perform their own shutdown processing.

### Usage Notes

1. The duration of the previous SHUTDOWN is reported only if the system has been restarted after a system failure or restarted via a SHUTDOWN REIPL command.
2. The QUERY SHUTDOWNTIME command returns the same information as the QUERY SHUTDOWN TIME command.

### Responses

Response 1:

```
System shutdown time: interval seconds
```

#### *interval*

is the amount of time reserved for VM SHUTDOWN processing.

Response 2:

```
System shutdown time: interval seconds; previous shutdown duration: length seconds
```

#### *interval*

is the amount of time reserved for VM SHUTDOWN processing.

#### *length*

is the amount of time the previous VM SHUTDOWN required.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*

## QUERY SIGNAL

---

►► Query — SIGnal — SHUTDOWNtime ◄◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY SIGNAL to display system default signal timeout interval value.

### Operands

#### SHUTDOWNtime

displays the system default shutdown signal timeout interval. If a SIGNAL, FORCE, or SHUTDOWN command is issued without specifying an interval, this value is used to determine the interval.

### Usage Notes

1. The system default signal timeout interval is set by the SET SIGNAL system configuration statement. The value can be dynamically reset for the current system IPL by the SET SIGNAL command.
2. A user is enabled for signals only when a guest operating system is running in the virtual machine and has enabled to receive these signals. To display the signal status of a user or all users, use the QUERY SIGNALS command.

### Responses

Response 1:

```
System default shutdown signal timeout: interval seconds
```

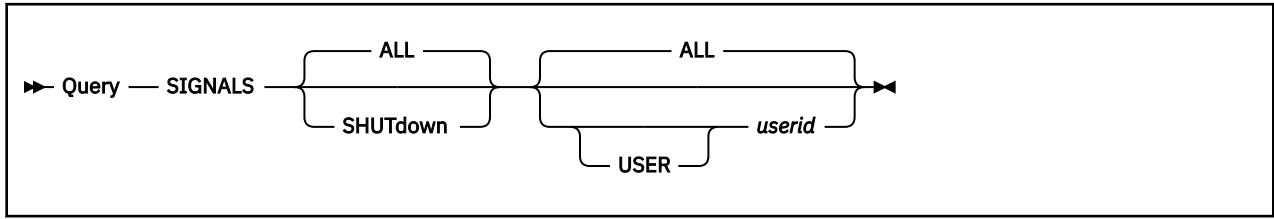
#### *interval*

is the duration of the system default shutdown signal timeout interval.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*

## QUERY SIGNALS



### Authorization

Privilege Class: B

### Purpose

Use QUERY SIGNALS to display a user's signal status or to display the signal status of all users who are enabled to receive them or are processing signals.

### Operands

#### ALL

indicates that the status of all signals is to be displayed.

#### SHUTDOWN

indicates that the status of the SHUTDOWN signal is to be displayed.

#### *userid*

identifies the user whose signal status is to be displayed. Use the keyword USER when you specify a user ID of "all".

#### ALL

indicates that the signal status of all users who are enabled to receive or have received the designated *signal* should be displayed.

### Responses

Response 1:

Userid	Signal	Signal Status	Signalled By	Timeout Remaining
<i>userid</i>	<i>signal</i>	Enabled	-	-
<i>userid</i>	<i>signal</i>	Pending	<i>issuer</i>	<i>interval</i>
<i>userid</i>	<i>signal</i>	Processing	<i>issuer</i>	<i>interval</i>
<i>userid</i>	<i>signal</i>	Not enabled	-	-
<i>userid</i>	-	Logging on/off	-	-
<i>userid</i>	-	Console wait	-	-

#### *userid*

identifies the user.

#### *signal*

identifies the signal.

#### Enabled

indicates that the virtual machine is enabled to receive the *signal*.

#### Pending

indicates that the virtual machine has the *signal* pending but it has not yet been delivered.

#### Processing

indicates that the virtual machine has had the *signal* delivered it but has not yet responded to indicate that it has completed processing it.



**Not enabled**

indicates that the virtual machine is not enabled to receive the *signal*.

**Logging on/off**

indicates that the virtual machine is in the process of logging on or off.

**Console wait**

indicates that the virtual machine's console is in a state (for example, HOLDING) such that the signal status cannot be obtained.

**issuer**

identifies the user who caused the signal to be sent to the user. If SYSTEM is specified as the issuer in the output response, the signal came from the hardware (a hardware quiesce signal, for example) rather than an actual user.

**interval**

shows the amount of time in seconds remaining in the timeout interval.

Response 2:

```
No users are enabled for signals
```

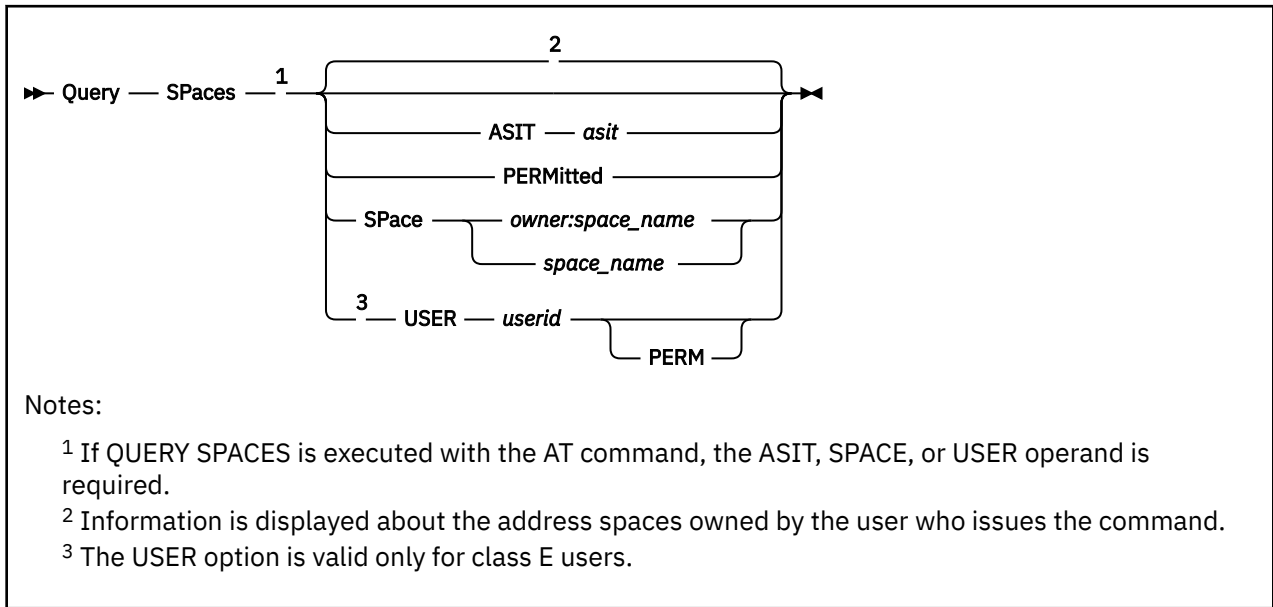
**Usage Notes**

1. A user is enabled for signals only when a guest operating system is running in the virtual machine and has enabled to receive these signals.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP013E Conflicting option - *option*
- HCP045E *userid* not logged on
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*

## QUERY SPACES



### Authorization

Privilege Class: E, G

### Purpose

Use QUERY SPACES to display information about the address spaces that a virtual machine owns or has access to. If you are a class G user, you can display information about your own address spaces. You can also display information about an address space owned by another user ID if you have permission to access it. If you are a class E user, you can display information about any address space, or all address spaces that a specific user ID owns or has access to.

### Operands

#### ASIT *asit*

returns information about the address space identified by the ASIT *asit*.

#### PERMitted

returns information about the address spaces that the command issuer has permission to access.

#### SPace *owner:space\_name*

returns information about the address space owned by user ID *owner* and named *space\_name*. The owner's user ID is optional and defaults to the ID of the user issuing the command. The space name is a 1- to 24-character string of alphanumeric and national characters.

The class G user must have permission to access the address space. The class E user can obtain the information without permission to access the space.

#### USER *userid*

returns information about the address spaces owned by *userid*.

#### PERM

returns information about the address spaces that *userid* has permission to access.

## Usage Notes

1. Programs that parse the output of this command should not depend on the ordering of the columns in the output. They should use the heading line to determine which column contains a particular piece of information.
2. If you issue QUERY SPACES with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the QUERY SPACES command is rejected and error message HCPQSP6578E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CClear option
  - SET MACHine (to a different machine architecture)

## Responses

Response 1:

```

ASIT      STORAGE    P/S      ACC      SPACE IDENTIFICATION
aaa...aaa nu      {PRV}    {R/O}    sss...sss
                {SHR}    {R/W}
                {---}
    
```

### **aaa...aaa**

is the address space identification token (ASIT) for the address space (16 hexadecimal digits).

### **nu**

where *n* is the size of the address space in bytes and *u* is the unit of measure:

#### **M**

megabytes

#### **G**

gigabytes

#### **T**

terabytes

#### **P**

petabytes

#### **E**

exabytes

For the primary address space, this includes discontinuous storage (that is - NSS or saved segment storage located above the virtual machine's defined size). If discontinuous storage is used, there may be non-addressable areas within this storage.

### **PRV**

indicates that the address space is in the private state.

### **SHR**

indicates that the address space is in the shareable state.

### **sss...sss**

is the space ID for the address space (3 to 33 characters).

### **R/O**

### **R/W**

### **---**

show the degree of access permitted to the user. Three dashes are displayed when a class E user uses the ASIT or SPACE operands for a space the user does not have permission to access.

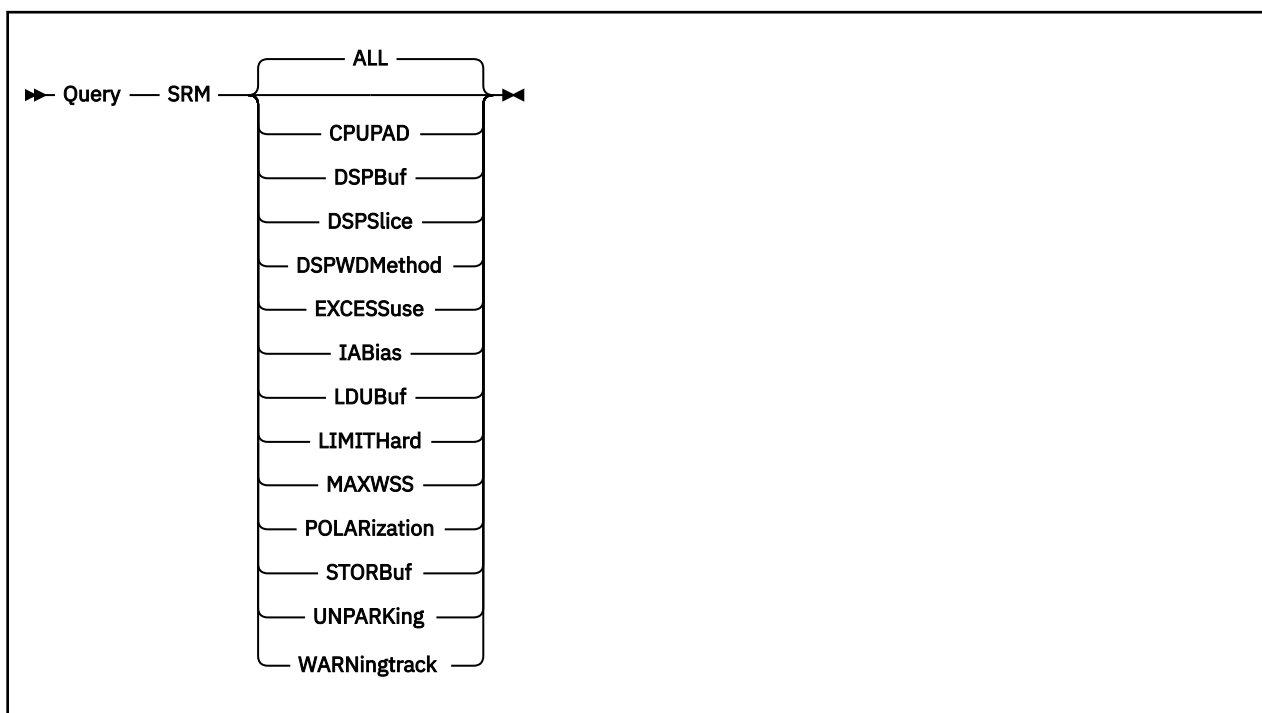
### Examples

```
q sp
ASIT          STORAGE P/S ACC SPACE IDENTIFICATION
017801800000001 32M PRV R/W OPERATOR:spacename
017802400000001  1M PRV R/W OPERATOR:spacename
017802800000001  1M PRV R/W OPERATOR:spacename
017802C00000001  1M PRV R/W OPERATOR:spacename
```

### Messages

- HCP003E Invalid option - *option*
- HCP007E Invalid userid - *userid*
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command.
- HCP1001E An operand is missing for *option*.
- HCP6156E {ASIT *asit*|Address space identifier SPACE\_ID} does not designate an address space to which you have the appropriate access.
- HCP6578E Invalid {access register|ASIT|address space identifier} - *option*

## QUERY SRM



### Authorization

Privilege Class: A, E

### Purpose

Use QUERY SRM (system resource manager) to display system-wide parameters used by the scheduler to set the priority of system resource access.

### Operands

#### ALL

causes a summary of the current system-wide settings. The operand displays the CPUPAD, DSPBUF, DSPSLICE, DSPWDMETHOD, EXCESSUSE, IABIAS, LDUBUF, LIMITHARD, MAXWSS, POLARIZATION, STORBUF, UNPARKING, and WARNINGTRACK settings. This operand is the default.

#### CPUPAD

displays the current CPUPAD setting for each CPU type. These settings control the amount of additional CPU capacity the system should be prepared to consume beyond what the system projects it will need. For additional details about this setting, see [“SET SRM” on page 1704](#).

For additional information on CPUPAD and HiperDispatch, see the [SRM Statement](#) and [z/VM HiperDispatch](#) in *z/VM: CP Planning and Administration*.

#### DSPBuf

displays the current DSPBUF (dispatch buffer) values used by the scheduler to limit the number of users allowed in the dispatch list for each transaction class. For a description of the values, see [“SET SRM” on page 1704](#).

#### DSPSlice

displays the current setting of the dispatching timeslice in milliseconds.

**DSPWDMethod**

displays the current DSPWDMethod setting. This indicates which work distribution algorithm the dispatcher is using.

For additional information on DSPWDMETHOD and HiperDispatch, see the [SRM Statement](#) and [z/VM HiperDispatch](#) in *z/VM: CP Planning and Administration*.

**EXCESSuse**

displays the current EXCESSUSE (unentitled CPU use) setting for each CPU type. These settings control how aggressively the system should attempt to use CPU capacity beyond its entitlement. For additional details about this setting, see [“SET SRM” on page 1704](#).

For additional information on EXCESSUSE and HiperDispatch, see the [SRM Statement](#) and [z/VM HiperDispatch](#) in *z/VM: CP Planning and Administration*.

**IABias**

specifies that the current settings for the interactive bias *intensity* and *duration* should be displayed. For a description of the interactive bias facility, see [“SET SRM” on page 1704](#).

**LDUBuf**

displays the current LDUBUF (loading user buffer) percentages used by the scheduler to partition access to the paging configuration based on transaction classes. For a description of the percentage values, see [“SET SRM” on page 1704](#).

**LIMITHard**

displays the method the scheduler uses to enforce hard limiting of scheduled system resources. This setting only affects users with absolute maximum shares defined with the SET SHARE command or the SHARE directory statement. The LIMITHARD *method* can be either CONSUMPTION or DEADLINE. For a description of these methods, see [“SET SRM” on page 1704](#).

**MAXWSS**

displays the current maximum working set size (MAXWSS) percentage of pageable real storage and the corresponding number of page frames.

**POLARization**

displays the partition's polarization mode and whether the partition has access to global performance data.

For additional information on POLARIZATION and HiperDispatch, see the [SRM Statement](#) and [z/VM HiperDispatch](#) in *z/VM: CP Planning and Administration*.

The display includes:

1. Polarization mode:
  - VERTICAL. The partition is running in vertical polarization mode. This mode allows logical CPUs in a partition to receive differing entitlements.
  - HORIZONTAL. The partition is running in horizontal polarization mode. Each online (configured) logical CPU of a particular CPU type gets an equal portion of the partition's weight for that CPU type.

For additional details about these settings, see [“SET SRM” on page 1704](#).
2. Global performance data indicator. Global performance data (GPD) is a setting turned on in the partition's activation profile on the HMC (Hardware Management Console) or SE (Support Element). This setting allows PR/SM to tell a partition about usage on other partitions within the CPC.

**STORBuf**

displays the current STORBUF (storage buffer) percentages used by the scheduler to partition storage based on transaction classes. For a description of the percentage values and the transaction classes represented, see [“SET SRM” on page 1704](#).

**UNPARKing**

displays the setting of the unparking heuristic.

**WARNingtrack**

displays the current warning-track setting, along with facility availability.

## Usage Notes

1. For explanations of all settings displayed by the QUERY SRM command, see “SET SRM” on page 1704.
2. For information about the allocation of storage resources to users, see [Checking the Allocation of Real Storage in z/VM: System Operation](#).

## Responses

Response 1:

```
LIMITHARD METHOD: method
IABIAS : INTENSITY=mmm%; DURATION=nnn
LDUBUF : Q1=ddd% Q2=eee% Q3=fff%
STORBUF: Q1=aaa% Q2=bbb% Q3=ccc%
DSPBUF : Q1=iiii Q2=jjjj Q3=kkkk
DISPATCHING MINOR TIMESLICE = xxx MS
MAXWSS : LIMIT=nnnn%
..... : PAGES=mmmmmmmmmm
XSTORE : ---
POLARIZATION: type
GLOBAL PERFORMANCE DATA: ON|OFF|NA
EXCESSUSE: CP-excess IFL-excess ICF-excess ZIIP-excess
CPUPAD: CP-qqqq% IFL-qqqq% ICF-qqqq% ZIIP-qqqq%
DSPWMETHOD: RESHUFFLE|REBALANCE
UNPARKING: LARGE|MEDIUM|SMALL
WARNING TRACK: AVAILABLE|UNAVAILABLE ON|OFF
```

is the response to QUERY SRM ALL, where *excess* = {HIGH|MEDIUM|LOW|NONE}.

Response 2:

```
LIMITHARD METHOD: method
```

is the response to QUERY SRM LIMITHARD.

Response 3:

```
IABIAS : INTENSITY=mmm%; DURATION=nnn
```

is the response to QUERY SRM IABIAS.

Response 4:

```
LDUBUF : Q1=ddd% Q2=eee% Q3=fff%
```

is the response to QUERY SRM LDUBUF.

Response 5:

```
STORBUF: Q1=aaa% Q2=bbb% Q3=ccc%
```

is the response to QUERY SRM STORBUF.

Response 6:

```
DSPBUF : Q1=iiii Q2=jjjj Q3=kkkk
```

is the response to QUERY SRM DSPBUF.

Response 7:

```
DISPATCHING MINOR TIMESLICE = xxx MS
```

is the response to QUERY SRM DSPSLICE.

Response 8:

## QUERY SRM

```
MAXWSS : LIMIT=nnnn%  
... .. : PAGES=mmmmmmmmmm
```

is the response to QUERY SRM MAXWSS.

Response 9:

```
POLARIZATION: type  
GLOBAL PERFORMANCE DATA: ON|OFF|NA
```

is the response to QUERY SRM POLARIZATION.

Response 10:

```
EXCESSUSE: CP-excess IFL-excess ICF-excess ZIIP-excess
```

is the response to QUERY SRM EXCESSUSE, where *excess* = { HIGH | MEDIUM | LOW | NONE }.

Response 11:

```
CPUPAD: CP-qqqqq% IFL-qqqqq% ICF-qqqqq% ZIIP-qqqqq%
```

is the response to QUERY SRM CPUPAD.

Response 12:

```
DSPWDMETHOD: RESHUFFLE|REBALANCE
```

is the response to QUERY SRM DSPWDMETHOD.

Response 13:

```
UNPARKING: { LARGE | MEDIUM | SMALL }
```

is the response to QUERY SRM UNPARKING.

Response 14:

```
WARNING TRACK: AVAILABLE|UNAVAILABLE ON|OFF
```

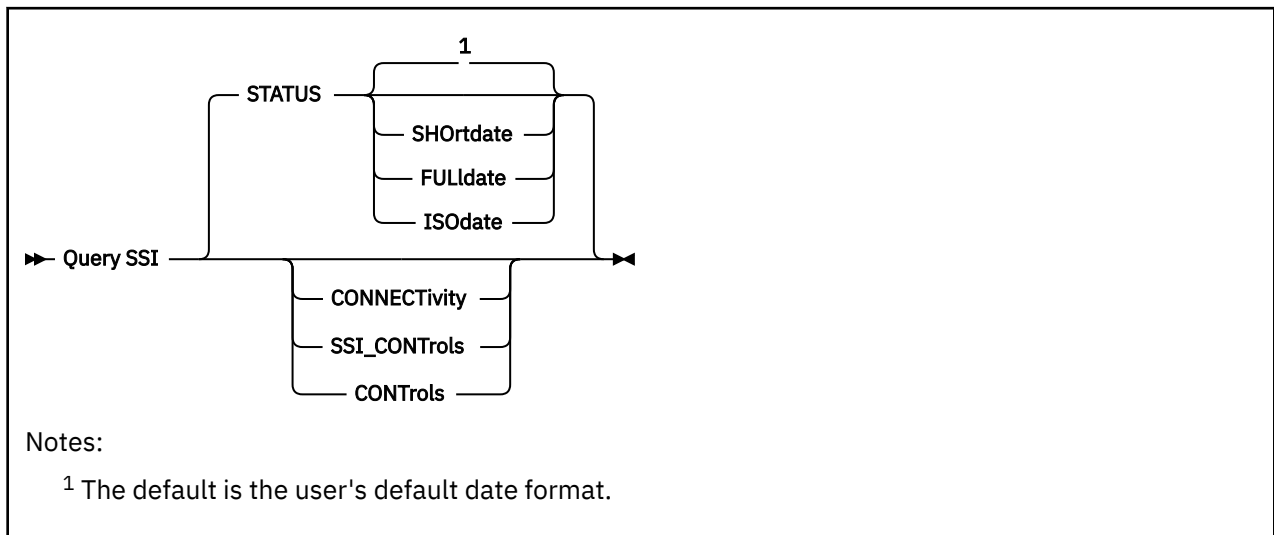
is the response to QUERY SRM WARNINGTRACK.

### Messages

- HCP026E Operand missing or invalid
- HCP1401I Expanded storage not supported.



## QUERY SSI



### Authorization

Privilege Class: B, E

### Purpose

Use QUERY SSI to display the single system image (SSI) name, member status, and connectivity status.

### Operands

#### STATUS

displays the state and heartbeat information for each of the SSI members.

#### CONNECTivity

displays the status of connections to the SSI members.

#### SSI\_CONTRols

#### CONTRols

displays the SPOOL\_MEMBERS setting of the SSI\_CONTROLS statement in the system configuration file for each of the joined SSI members. This setting determines among how many members of the SSI cluster the spool file ID numbers are allocated for use when new spool files are created. For more information about the SSI\_CONTROLS statement, see *z/VM: CP Planning and Administration*.

#### SHOrtdate

specifies that dates in the response to this command are displayed in *mm/dd/yy* format, where *mm* is the month, *dd* is the day of the month, and *yy* is the 2-digit year.

#### FULldate

specifies that dates in the response to this command are displayed in *mm/dd/yyyy* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

#### ISOdate

specifies that dates in the response to this command are displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

### Usage Notes

1. You can use the SET SSI command to add systems to available slots in the SSI member list or to remove members from the SSI member list.

2. The default date format for certain CP and CMS commands can be set on a system-wide basis and also for the individual user. The system-wide default date format is set with the SYSTEM\_DATEFORMAT system configuration statement. The user's default date format is set with the DATEFORMAT directory statement. The system-wide default and the user's default can also be set with the CP SET DATEFORMAT command. The user's default date format defaults to the system-wide default. The system-wide and user settings can be queried with the CP QUERY DATEFORMAT command. The hierarchy of possible date format settings for the QUERY SSI command, from highest priority to lowest, is:
  - QUERY SSI operand
  - User default
  - System-wide default.
3. When the response indicates that a slot is AVAILABLE, it means there is no member defined in that slot, but the slot is supported at this level of z/VM. It might be possible to use the SET SSI SLOT command to define a member in that slot, but this is not always the case. Whether those slots are usable or not depends on the settings of the SPOOL\_MEMBERS operand of the SSI\_CONTROLS statement in the system configuration file for all joined members of the cluster. For example, a member can only be defined in slots 5-8 if all joined members are running with a SPOOL\_MEMBERS setting of 8 rather than 4.

## Responses

1. If you enter any form of the QUERY SSI command on a system that is not a member of an SSI cluster, you might see a response similar to this:

```
query ssi status
This system is not a member of an SSI cluster
```

2. If you enter the QUERY SSI STATUS command on a system that is a member of an SSI cluster, you might see a response similar to this:

```
query ssi
SSI Name:  ALEAST
SSI Mode:  Stable
Cross-System Timeouts: Enabled
SSI Persistent Data Record (PDR) device: SSIVOL on 3333
SLOT SYSTEMID STATE      PDR HEARTBEAT      RECEIVED HEARTBEAT
  1 YANKEES  Joined    01/30/2010 12:03:33 01/30/2010 12:03:35
  2 REDSOX   Joined    01/30/2010 12:03:47 01/30/2010 12:03:49
  3 ORIOLES  Down (shut down successfully)
  4 ----- Available
  5 ----- Available
  6 ----- Available
  7 ----- Available
  8 ----- Available
```

where:

**SSI Name: ALEAST**

ALEAST is the name of the SSI cluster.

**SSI Mode: Stable**

**SSI Mode: Influx**

**SSI Mode: Safe**

indicates this member's view of the SSI mode. Valid SSI modes are:

**Stable**

All IPLed systems in the SSI cluster are joined and participating in all SSI services.

**Influx**

A member is in a transition state.

**Safe**

A member is in an unknown state.

**Cross-System Timeouts: Enabled | Disabled**

indicates the status of cross-system timeouts. For more information, refer to the CROSS\_SYSTEM\_TIMEOUTS operand on the FEATURES configuration statement. See [FEATURES Statement](#) in *z/VM: CP Planning and Administration*.

**SSI Persistent Data Record Device (PDR): SSIVOL on 3333**

SSIVOL is the volume label and 3333 is the real device address of the device that contains the SSI persistent data record.

**SLOT**

indicates the slot number in the SSI member list.

**SYSTEMID**

is the system name of the member occupying that slot.

**STATE**

is the issuing member's view of the state of the associated member. Valid states are:

**Down**

A member is in the Down state when any of the following are true, as indicated by the parenthetical text that follows:

- (shut down successfully) it has left the SSI cluster due to a system shutdown.
- (not shut down successfully) it has left the SSI cluster due to a system abend.
- (declared down) it has been declared down by use of the SET SSI command.
- (not IPLed) it has not been IPLed as a member of the SSI cluster.
- (initializing, not yet joining) it has been IPLed, but it hasn't joined the SSI cluster.
- (reason undetermined)
- (reason not valid)

**Joining**

A member is in the Joining state when it is in the process of joining an SSI cluster that already has one or more joined members. Only one member can be in the Joining state at a time.

**Joined**

A member is in the Joined state when it has successfully joined the SSI cluster and is participating in SSI-wide operations.

**Leaving**

A member is in the Leaving state when it was joined to an SSI cluster and is shutting down.

**Isolated**

A member is in the Isolated state when it cannot join the SSI cluster due to a failure in the enablement of SSI-wide operations or due to a failure occurring while attempting to join the SSI cluster.

**Suspended**

A member is in the Suspended state when it does not have connectivity to another member in the SSI cluster that is in a state other than Down or Isolated.

**Unknown**

A remote member is in the Unknown state when the connectivity from the local member is lost.

**PDR HEARTBEAT**

is the timestamp in the local member's time zone of the heartbeat in the SSI persistent data record for the specified member. Besides containing a timestamp, the PDR heartbeat field can contain one of the following values:

- Error reading PDR
- System not found
- Hyperswap occurring

**RECEIVED HEARTBEAT**

is the timestamp in the local member's time zone of the last received heartbeat from the specified member.

3. If you enter the QUERY SSI command on a system that has been IPLed with the REPAIR IPL parameter, you might see a response similar to the following:

```
query ssi status
This system was IPLed with the REPAIR IPL parameter and is
currently operating as if not in an SSI cluster.
```

4. If you enter the QUERY SSI CONNECTIVITY command on system YANKEES, you might see a response similar to this:

```
q ssi connect
SSI Name: ALEAST
SSI Mode: Stable
Cross-System Timeouts: Disabled
SLOT SYSTEMID STATE CONNECT TYPE HOPS
 1 YANKEES Joined Local -
 2 REDSOX Joined Direct 0
 3 ORIOLES Joined Direct 0
 4 ----- Available
 5 ----- Available
 6 ----- Available
 7 ----- Available
 8 ----- Available
```

where:

**SSI Name: ALEAST**

ALEAST is the name of the SSI cluster.

**SSI Mode: Stable**

**SSI Mode: Influx**

**SSI Mode: Safe**

indicates this member's view of the SSI mode. Valid SSI modes are:

**Stable**

All IPLed systems in the SSI cluster are joined and participating in all SSI services.

**Influx**

A member is in a transition state.

**Safe**

A member is in an unknown state.

**Cross-System Timeouts: Enabled | Disabled**

indicates the status of cross-system timeouts. For more information, refer to the CROSS\_SYSTEM\_TIMEOUTS operand on the FEATURES configuration statement. See the [FEATURES Statement in z/VM: CP Planning and Administration](#).

**SLOT**

indicates the slot number in the SSI member list.

**SYSTEMID**

is the system name of the member occupying that slot.

**STATE**

is the issuing member's view of the state of the associated member. Valid states are:

**Down**

A member is in the Down state when any of the following are true, as indicated by the parenthetical text that follows:

- (shut down successfully) it has left the SSI cluster due to a system shutdown.
- (not shut down successfully) it has left the SSI cluster due to a system abend.
- (declared down) it has been declared down by use of the SET SSI command.

- (not IPLed) it has not been IPLed as a member of the SSI cluster.
- (initializing, not yet joining) it has been IPLed, but it hasn't joined the SSI cluster.
- (reason undetermined)
- (reason not valid)

**Joining**

A member is in Joining state when it is in the process of joining an SSI cluster that already has one or more joined members. Only one member can be in the Joining state at a time.

**Joined**

A member is in the Joined state when it has successfully joined the SSI cluster and is participating in SSI-wide operations.

**Leaving**

A member is in the Leaving state when it was joined to an SSI cluster and is shutting down.

**Isolated**

A member is in the Isolated state when it cannot join the SSI cluster due to a failure in the enablement of SSI-wide operations or due to a failure occurring while attempting to join the SSI cluster.

**Suspended**

A member is in the Suspended state when it does not have connectivity to another member in the SSI cluster that is in a state other than Down or Isolated.

**Unknown**

A remote member is in the Unknown state when the connectivity from the local member is lost.

**CONNECT TYPE**

is the type of connection the issuing member has to the associated system. Values for CONNECT TYPE include:

**Not connected**

There is no ISFC connection to this system.

**Direct**

There is a direct ISFC connection to this system.

**Local**

This is the system on which the command was issued.

**Incomplete**

The process to determine connectivity is not complete. Reissue the command to see the connection status.

**HOPS**

Specifies the number of hops to the node. This field has a value of 0 or –.

5. If you enter the QUERY SSI SSI\_CONTROLS command, you might see a response similar to this:

```
q ssi ssi_controls
SSI Name: ALEAST
SSI Mode: Stable
Cross-System Timeouts: Disabled
SLOT SYSTEMID STATE      SPLMEMS
  1 YANKEES  Joined      8
  2 REDSOX   Down        -
  3 ORIOLES  Joined      4
  4 ----- Available
  5 ----- Available
  6 ----- Available
  7 ----- Available
  8 ----- Available
```

where:

All of the response fields are defined in response “2” on page 1178, except the following field, which is unique to the SSI\_CONTROLS response:

## QUERY SSI

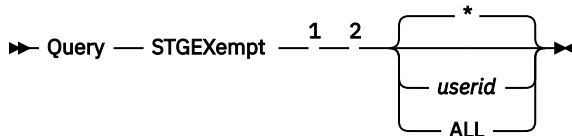
### SPLMEMS

shows the spool members setting of each joined member of the SSI cluster. This indicates the spool ID assignment algorithm in use by the member as set on the SSI\_CONTROLS statement in the system configuration file. A hyphen appears for any member that is currently not joined.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*

## QUERY STGEXEMPT



### Notes:

- <sup>1</sup> Operands are not valid for class G users.
- <sup>2</sup> If QUERY STGEXEMPT is executed with the AT command, the *userid* or ALL operand is required.

### Authorization

Privilege Class: A, B, C, E, G

### Purpose

Use QUERY STGEXEMPT to determine the state of free storage limit detection for a virtual machine. A class G user can display the state of free storage limit detection for the user's own virtual machine. A class A, B, C, or E user can display the state of free storage limit detection for any user or all users.

### Operands

**\***

displays the state of free storage limit detection for the user who issues the command.

**userid**

specifies the user ID for which the state of free storage limit detection is to be displayed.

**ALL**

specifies that the state of free storage limit detection for all logged on users is to be displayed.

### Usage Notes

1. The QUERY STGEXEMPT command displays the state of free storage limit detection for the user ID. If a user ID is exempt from free storage limit detection, usage of CP storage in support of the user ID is unbounded. If a user ID is subject to free storage limit detection, CP will detect if too much CP storage is consumed in support of the user ID and if so, the user ID will be stopped and possibly forced off the system.

### Responses

Response 1:

```
Userid USERID is {exempt from} free storage limit detection.
                  {subject to }
```

is the response for QUERY STGEXEMPT, QUERY STGEXEMPT USERID, or QUERY STGEXEMPT \*.

There may be additional response saying that free storage limit detection is off for the system.

Response 2:

```
The following users are exempt from
free storage limit detection:
```

## QUERY STGEXEMPT

```
USERID1 USERID2 USERID3 USERID4 USERID5  
USERID6 USERID7 USERID8 USERID9 ...
```

is the response for QUERY STGEXEMPT ALL.

Response 3:

```
There are no users that are exempt  
from free storage limit detection
```

is the response for QUERY STGEXEMPT ALL when no users are exempt.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP0045E *userid* not logged on



# QUERY STGLIMIT

---

► Query — STGLimit ◄

## Authorization

Privilege Class: A, B, C, E

## Purpose

Use QUERY STGLIMIT to determine the state of free storage limit detection for the system.

## Usage Notes

1. If free storage limit detection for the system is off, usage of CP storage in support of the user ID is unbounded.
2. If free storage limit detection for the system is on, CP will detect if too much CP storage is consumed in support of userids and if so, those userids consuming excessive free storage will be stopped and possibly forced off the system.

## Responses

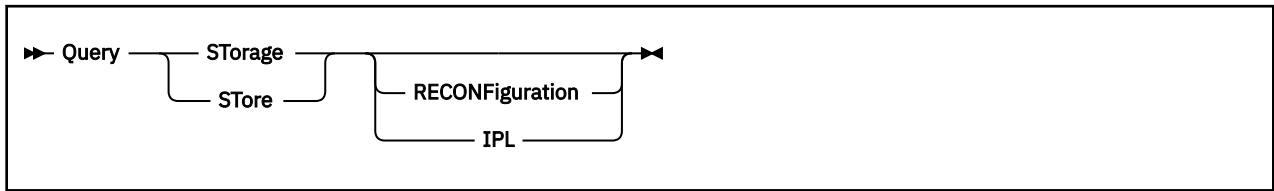
Response 1:

```
FREE STORAGE LIMIT DETECTION IS {ON }  
                                {OFF}
```

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

## QUERY STORAGE/STORE



### Authorization

Privilege Class: A, B, E

### Purpose

Use QUERY STORAGE or QUERY STORE to display z/VM's real storage configuration. The command also displays the status of a storage reconfiguration in progress, and the values used to initialize the storage configuration the last time the system was started.

### Operands

#### RECONFiguration

displays the status of a storage reconfiguration in progress. If a reconfiguration is not in progress, the current storage configuration is displayed.

#### IPL

displays the current amount of permanent and reconfigurable storage, along with the storage specification that z/VM used the last time the system was started.

### Usage Notes

1. When you define a logical partition, two storage allocations, *Initial* and *Reserved*, are specified in the image profile for the partition. The maximum amount of storage that can be used by an operating system running in the partition is the sum of these two storage allocations. The machine uses the *Initial* and *Reserved* amounts in determining the storage-increment size. The storage-increment size is the unit in which storage can be added to or removed from the configuration.

For a fresh IPL (one that is not an automatic restart), the following considerations apply:

- CP uses the PERMANENT and RECONFIGURABLE specifications on the STORAGE statements in the system configuration file to determine how to initialize storage. There must be at least 4 GB of permanent storage online before any reconfigurable storage can be configured.
- Alternatively, you can use the IPL parameter STORE=*n* to override the storage configuration information in the system configuration file. In this case, all of the storage specified via STORE=*n* is initialized as permanent.
- If neither the STORAGE statement nor the IPL parameter STORE= was specified, all of the storage that z/VM finds configured in the partition is initialized as permanent storage. If this is the first IPL after the logical partition was activated, the amount of configured storage equals the *Initial* storage allocation. If this is a subsequent IPL, the amount of storage configured at IPL depends on the configuration left by the operating system previously running in the partition.

The storage values on the STORAGE statement and the IPL parameter STORE=*n* are rounded up to a multiple of the storage-increment size, if necessary.

For an IPL that is the result of an automatic restart, the AFTER\_RESTART and AFTER\_SHUTDOWN\_REIPL operands on the STORAGE statement in the system configuration file determine z/VM's behavior in establishing the storage configuration. The AFTER\_RESTART operand applies to an automatic reIPL resulting from a CP abend or a PSW restart. The

AFTER\_SHUTDOWN\_REIPL operand applies to an automatic reIPL resulting from the SHUTDOWN command with the REIPL operand. Parameters on these operands provide the ability for you to either initialize the storage configuration as though it were a fresh IPL, or to keep the storage configuration as it was at system termination prior to the reIPL. See the [STORAGE Statement](#) in *z/VM: CP Planning and Administration* for more information.

2. Usable storage is not necessarily contiguous. There can be unaddressable address ranges resulting from a SET STORAGE command to remove reconfigurable storage.
3. The STORAGE= and CONFIGURED= values in the QUERY STORAGE command response might include frames taken offline due to an error. The OFFLINE= value displayed in the QUERY FRAMES command response is the count of frames taken offline due to an error.
4. The sum of CONFIGURED=, STANDBY=, and RESERVED equals the sum of the Initial and Reserved storage allocations in the image profile for the logical partition, or the z/VM-supported maximum, whichever is less.
5. The STORAGE=, CONFIGURED=, and Reconfigurable= values might continue to change after a storage reconfiguration completes. When storage is removed from the reconfiguration, it is done in two parts: vacating the storage and transitioning the storage to the standby state. After the storage is completely vacated, it might take extra time to complete the storage transitions. The SET STORAGE command will have already returned, but the values on the QUERY STORAGE command might continue to update until the transitions are complete.
6. A VIRTUAL version of this command, QUERY VIRTUAL STORAGE/STORE, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL STORAGE/STORE”](#) on page 1333.

## Responses

Response 1:

When a QUERY STORAGE command is issued and a storage reconfiguration is in progress to add storage to the z/VM configuration, you will receive the following command response:

```
STORAGE = nu CONFIGURED = nu INC = nu STANDBY = nu RESERVED = nu
Permanent = nu Reconfigurable = nu Maximum STORAGE = nu
```

where:

### **STORAGE = nu**

specifies the size of real storage (permanent plus reconfigurable) in use by z/VM).

### **CONFIGURED = nu**

specifies the amount of real storage configured for the logical partition in which z/VM is running.

### **INC = nu**

specifies the size of the real storage increment. Storage must be added or removed in multiples of the increment size.

### **STANDBY = nu**

specifies the amount of real storage in standby state that is available to be brought online with the SET STORAGE command. Standby storage is a calculated value based on the amount of installed storage that is not currently claimed by active logical partitions and the reserved storage specified for the logical partition in which z/VM is running, subject to the z/VM-supported maximum.

### **RESERVED = nu**

specifies the amount of real storage in reserved state that can become available via manual storage actions (if another logical partition is deactivated, for example).

### **Permanent = nu**

specifies the amount of online real storage that cannot be removed.

**Reconfigurable = nu**

specifies the amount of online real storage that can be removed using the SET STORAGE command. If your z/VM system is running on a machine that does not support reconfigurable storage, the reconfigurable storage value is displayed as Reconfigurable = --.

**Maximum STORAGE = nu**

specifies the largest amount of storage that can be brought online to z/VM in this partition. The value is the sum of the Initial + Reserved settings for the logical partition in the image profile, or is the maximum amount of real storage this z/VM system supports, whichever is less.

**u**

is the storage unit suffix:

**M**

megabytes

**G**

gigabytes

**T**

terabytes

**P**

petabytes

**E**

exabytes

Response 2:

When a QUERY STORAGE RECONFIGURATION command is issued and a storage reconfiguration is in progress to add storage to the z/VM configuration, you will receive the following response:

```
STORAGE = nu CONFIGURED = nu INC = nu STANDBY = nu RESERVED = nu
Permanent = nu Reconfigurable = nu Maximum STORAGE = nu
Storage increase in progress. Elapsed time = ddd hh:mm:ss
Target:          {Permanent = nu} {Reconfigurable = nu}
Total to add:   {Permanent = nu} {Reconfigurable = nu}
Remainder to add: {Permanent = nu} {Reconfigurable = nu}
Storage reconfiguration is active.
```

The first two lines of output are described under Response 1. The values on the additional lines are as follows:

**Elapsed time = ddd hh:mm:ss**

specifies the amount of time in days, hours, minutes, and seconds that the add operation has been in progress.

**Target: {Permanent = nu} {Reconfigurable = nu}**

specifies the amount of permanent storage, reconfigurable storage, or both that will be in use by z/VM after the add operation completes.

**Total to add: {Permanent = nu} {Reconfigurable = nu}**

specifies the amount of permanent storage, reconfigurable storage, or both to be added as requested on the SET STORAGE command that initiated the reconfiguration.

**Remainder to add: {Permanent = nu} {Reconfigurable = nu}**

specifies the amount of permanent storage, reconfigurable storage, or both that still needs to be added for the storage reconfiguration in progress.

**Storage reconfiguration is active.**

indicates that a reconfiguration is in progress. This will appear as the last line of output if a user issued a SET STORAGE command, or if the system is initializing and needs to initialize or remove storage to match the requested configuration. SET STORAGE commands cannot be issued while another storage reconfiguration is in progress.

Response 3:

When a QUERY STORAGE RECONFIGURATION command is issued and a storage reconfiguration is in progress to remove storage from the z/VM configuration, you will receive the following response:

```
STORAGE = nu CONFIGURED = nu INC = nu STANDBY = nu RESERVED = nu
Permanent = nu Reconfigurable = nu Maximum STORAGE = nu
Storage decrease in progress. Elapsed time = ddd hh:mm:ss
Target Reconfigurable = nu
Total to remove = nu
Remainder to remove = nu
MAXPAGEFULL = nnn% | Forced to no limit
Storage reconfiguration is active.
```

The first two lines of output are described under Response 1. The values on the additional lines are as follows:

**Elapsed time = ddd hh:mm:ss**

specifies the amount of time in days, hours, minutes, and seconds that the removal operation has been in progress.

**Target reconfigurable = nu**

specifies the amount of reconfigurable storage that will be in use by z/VM after the removal operation completes.

**Total to remove = nu**

specifies the amount of reconfigurable storage to be removed as requested on the SET STORAGE command initiating the reconfiguration.

**Remainder to remove = nu**

specifies the amount of reconfigurable storage that still needs to be removed for the storage reconfiguration in progress.

**MAXPAGEFULL = nnn%**

is the maximum percent of paging slots in use that you will tolerate for this reclamation. This is the value of the MAXPAGEFULL operand specified on the SET STORAGE command, or if MAXPAGEFULL was not specified on the command, it is the PAGING WARNING value specified via the PAGING system configuration statement or a previous SET PAGING command. If the amount of paging space in use reaches this percentage, the storage removal operation terminates with message HCP2650E. Any storage already removed is not brought back online.

When `Forced to no limit` is displayed, the `FORCE` operand was specified on the SET STORAGE remove command, causing all viability checks to be bypassed.

Response 4:

When a QUERY STORAGE IPL command is issued and a storage reconfiguration is in progress to add storage to the z/VM configuration, you will receive the following response:

```
STORAGE = nu CONFIGURED = nu INC = nu STANDBY = nu RESERVED = nu
Permanent = nu Reconfigurable = nu Maximum STORAGE = nu
IPL actual: Permanent = nu Reconfigurable = nu
IPL requested: Permanent = nu Reconfigurable = nu
IPL requested data source: specification
Last start was a type
```

The first two lines of output are described under Response 1. The values on the additional lines are as follows:

**IPL actual: Permanent = nu Reconfigurable = nu**

identifies the amount of permanent and reconfigurable storage brought online at the last start of the system. These values could differ from the requested amounts because CP rounded the values to a multiple of the storage-increment size, there is insufficient standby storage in the partition, or an error was encountered. Refer to the console log during IPL to determine if any storage configuration messages were issued. If your z/VM system is running on a machine that does not support reconfigurable storage, the reconfigurable storage value is displayed as `Reconfigurable = --`.

**IPL requested: Permanent = *nu* Reconfigurable = *nu***

identifies the amount of permanent and reconfigurable storage requested at the last start of the system.

**IPL requested data source: *specification***

identifies where the requested storage configuration specification was obtained the last time the system started. Possible values for *specification* are:

**IPL parameter STORE=*n***

storage configuration information for the last start was obtained from the IPL parameter STORE=*n*.

**IPL parameter STORE=INITIAL**

storage configuration information for the last start was obtained from the Initial storage allocation value in the image profile for the logical partition.

**STORAGE system configuration statement(s)**

storage configuration information for the last start was obtained from the PERMANENT specification, the RECONFIGURABLE specification, or both on the STORAGE statements in the system configuration file.

**None**

no explicit storage configuration data was provided on the IPL parameter STORE= or the STORAGE statement in the system configuration file. In this case, CP brings online only the amount of storage already configured to the partition and all of this storage is permanent.

**Kept from previous IPL**

the last start was an automatic restart. The STORAGE AFTER\_RESTART or AFTER\_SHUTDOWN\_REIPL statement in the system configuration file indicated that the storage configuration at system termination should be reestablished during the restart.

**Last start type was a *type***

identifies how the system was started. Possible values for *type* are:

**system IPL**

indicates that CP was started by the operator at the system console.

**system restart from SHUTDOWN REIPL**

indicates that the operator restarted CP using the CP SHUTDOWN command with the REIPL operand.

**system restart after system failure *XXXnnn***

indicates that CP restarted itself after an abend. *XXXnnn* is the abend code.

## Messages

- HCP003E Invalid Option - command contains extra option(s) starting with *operand*

## QUERY STP

---

```
▶▶ Query — STP ▶▶
```

### Authorization

Privilege Class: B

### Purpose

Use QUERY STP to display Server Time Protocol status.

### Responses

Response 1:

If you issue the QUERY STP command and the Server Time Protocol facility has not been installed on your machine, you will receive the following response:

```
Server Time Protocol facility not installed
```

Response 2:

If you issue the QUERY STP command and the Server Time Protocol facility is installed but not enabled for use, you will receive the following response:

```
Server Time Protocol facility not enabled
```

Response 3:

If the STP facility is installed and enabled, but STP has not been enabled from the system configuration file, you will receive the following response:

```
Server Time Protocol available but not enabled for CP use
```

Response 4:

If STP is enabled but CP has been unable to synchronize with the STP server and XRC\_OPTIONAL has not been specified in the system configuration file, you will receive the following response:

```
Server Time Protocol activation has not completed.  
I/O to XRC-capable DASD will be delayed until activation completes.
```

Response 5:

If STP is enabled but CP has been unable to synchronize with the STP server but XRC\_OPTIONAL was specified in the system configuration file, you will receive the following response:

```
Server Time Protocol activation has not completed.  
I/O to XRC-capable DASD will be issued without timestamps until  
activation completes.
```

Response 6:

If STP\_Timestamping is enabled and CP has successfully synchronized with STP, you will receive the following response:

```
Server Time Protocol synchronization activated for timestamping
```

## QUERY STP

### Response 7:

If QUERY STP is issued while VM is responding to an STP synchronization check, and XRC\_OPTional was not specified in the system configuration, you will receive the following response:

```
Server Time Protocol synchronization suspended.  
I/O to XRC-capable DASD will be delayed until synchronization completes.
```

### Response 8:

If QUERY STP is issued while VM is responding to an STP synchronization check, but XRC\_OPTional was specified in the system configuration, you will receive the following response:

```
Server Time Protocol synchronization suspended.  
I/O to XRC-capable DASD will be issued without timestamps until  
synchronization completes.
```

### Response 9:

If XRC\_TEST has been specified, you will receive the following response:

```
Server Time Protocol disabled: Timestamping enabled without  
a synchronized clock source for testing purposes.
```

### Response 10:

If STP has been enabled without timestamping support (for example, with the STP\_TZ statement) and CP has successfully synchronized with STP, you will receive the following response:

```
Server Time Protocol synchronization activated.
```

### Response 11:

If STP has been enabled without timestamping support (for example, with the STP\_TZ statement) and CP is currently responding to an STP synchronization check, you will receive the following response:

```
Server Time Protocol synchronization suspended.
```

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP0985E STP feature enabled but STP facility not installed
- HCP0985E STP feature enabled but STP facility not enabled
- HCP0985E Error *error* obtaining STP information
- HCP0985E STP Timing Mode not active
- HCP0985E STP not usable
- HCP0985E STP leap seconds not valid
- HCP0985E STP time zone offset not valid
- HCP0985E STP total time offset not valid
- HCP0985E Error *error* storing STP time zone information
- HCP0985E Error *error* setting STP controls
- HCP0986I TOD Clock synchronized via STP
- HCP0987E SET TIMEZONE not valid - STP timing mode active
- HCP0988I TOD Clock synchronization suspended



# QUERY SUBSTITUTE



## Authorization

Privilege Class: B

## Purpose

Use QUERY SUBSTITUTE to display the current virtual machine substitute definitions.

## Operands

### *alias*

is the alias definition to be displayed.

### ALL

causes all alias definitions to be displayed. If no operand is specified, ALL is assumed.

## Responses

Response 1:

Substitute	Userid	Defined By	Status
xxxxxxx	yyyyyyy	zzzzzzz	logged on
aaaaaaa	bbbbbbb	ccccccc	not logged on

Response 2:

*alias* is not a substitute

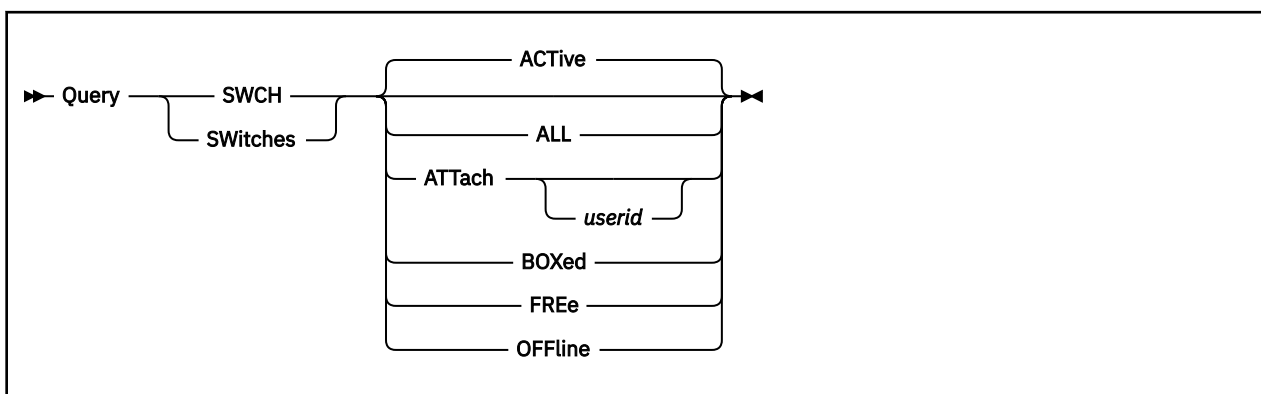
Response 3:

No substitutes defined or active

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*

## QUERY SWITCHES



### Authorization

Privilege Class: B

### Purpose

Use QUERY SWITCHES to display the status of real dynamic switching devices.

### Operands

#### ACTive

indicates active dynamic switching devices. An active device is in use by a user or the system and is specified as neither FREE nor OFFLINE.

#### ALL

indicates all dynamic switching devices attached to the system. The status is displayed in ACTIVE, FREE, and OFFLINE order and is equivalent to the response you would receive if you entered all of the following commands:

```
query swch active
query swch free
query swch offline
```

#### ATTach *userid*

indicates the dynamic switching devices that are dedicated to a user on the system. If you specify *userid*, only switching devices attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all switching devices dedicated to any user on the system. Attached devices are always active.

#### BOXed

indicates the dynamic switching devices that are unavailable, usually because of a hardware condition. A boxed device may also be active, free, or offline.

#### FREe

indicates all free dynamic switching devices. A free device is one not currently in use by a user or the system, and specified as neither ACTIVE nor OFFLINE.

#### OFFline

indicates all offline dynamic switching devices. An offline device is one specified as neither ACTIVE nor FREE.

## Usage Notes

1. A VIRTUAL version of this command, QUERY VIRTUAL SWITCHES, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL SWITCHES”](#) on page 1336.

## Responses

Response 1:

If you enter a QUERY SWCH, QUERY SWCH ACTIVE, or QUERY SWCH ATTACH command, and there is a dynamic switching device dedicated to a user, you see something like this:

```
q switches
SWCH 0CD5 ATTACHED TO EMILY 0012
Ready;
```

This means that the switching device at the real device number 0CD5 is attached to EMILY's virtual machine as the virtual device number 0012. ATTACHED may be replaced by BOX/ATTC, which indicates that the dedicated switching device is not available, usually because of a hardware condition.

Response 2:

If you enter a QUERY SWCH FREE command, and there is a free switching device attached to the system, you see something like this:

```
q swch free
SWCH 0387 FREE      , SWCH 03A1 FREE      , SWCH 03A2 FREE      , SWCH 03A3 FREE
SWCH 03A4 FREE      , SWCH 03A5 FREE      , SWCH 03A9 FREE      , SWCH 03AA FREE
Ready;
```

This response shows the real device number for each free switching device.

If you enter a QUERY SWCH OFFLINE or QUERY SWCH BOXED command, the response is in the same format as the response for QUERY SWCH FREE.

Response 3:

If you enter a QUERY SWCH ALL command, CP displays what it would display if you had entered QUERY SWCH ACTIVE, QUERY SWCH FREE, and QUERY SWCH OFFLINE commands.

Response 4:

If CP cannot find a switching device of the type you specified, you see a message in this format:

```
q switches boxed
A BOXED SWCH was not found.
Ready;
```

In this response, BOXED may be replaced with active, attached, free, or offline, depending on what you specified on the command.

## Messages

- HCP003E Invalid option - *option*
- HCP020E Userid missing or invalid
- HCP263E Too many operands were supplied for this command.

## QUERY SXSPAGES

► Query — SXSPages ◄

### Authorization

Privilege Class: A, B, E

### Purpose

Use QUERY SXSPAGES to display the status of pages in the system execution space below 2 GB (the system execution area). All system execution space pages below 2 GB are:

- Backed or unbacked – Backed pages have an associated frame in real storage; unbacked pages do not.
- Available or in-use – Available pages are candidates for usage by CP; in-use pages are used for aliases or free storage requests. An alias is a system execution space page associated with a pageable frame table entry.

Page usage for the system execution space above 2 GB (the frame table and the system execution space page management table) is not included in the response from this command.

If no real storage is available above 2 GB, those fields are still included in the response, but the status is displayed as zeros.

To display the host logical storage address ranges that are in use or usable by the system for the system execution space, use the QUERY SXSSTORAGE command.

### Usage Notes

1. The numbers displayed under Locked SXS pages identify only locked pages containing guest data.
2. Due to serialization, the number of free storage pages displayed under SXS pages in use might be different from the number of pages in the Free Storage section.
3. Trace pages associated with trace frames are in system execution space (SXS) and backed by a below-2-GB frame. This means that there is a limit on the number of trace frames due to the storage limit on SXS. An "insufficient storage" condition might occur when requesting more trace frames for each logical processor and will become more likely as the number of running logical processors in a configuration increases.

### Responses

All SXS pages:

Total=*n...n* Backed<2G=*n...n* Backed>2G=*n...n* Unbacked=*n...n*

Available SXS pages:

Total=*n...n* Backed<2G=*n...n* Backed>2G=*n...n* Unbacked=*n...n*

SXS pages in use:

Total=*n...n* Alias=*n...n* Nucleus/Prefix=*n...n* FreeStorage=*n...n* Other=*n...n*

Free Storage:

Backed<2G=*n...n* Backed>2G=*n...n* ReservedPages=*n...n*

Locked SXS pages:

Total=*n...n* ByCommand=*n...n*

Deferred SXS page requests waiting for:

Available page:

Alias=*n...n* Backed<2G=*n...n* BackedAny=*n...n*

Available frame:

```
Frame<2G=n...n  FrameAny=n...n
Total=n...n
```

All SXS pages:

**Total**

identifies the total number of pages in the system execution space:

```
Total = Backed<2G + Backed>2G + Unbacked
```

**Backed<2G**

identifies the number of pages in the system execution space that are backed with real storage frames below 2 GB.

**Backed>2G**

identifies the number of pages in the system execution space that are backed with real storage frames above 2 GB.

**Unbacked**

identifies the number of pages in the system execution space that are not backed.

Available SXS pages:

**Total**

identifies the total number of available pages in the system execution space:

```
Total = Backed<2G + Backed>2G + Unbacked
```

**Backed<2G**

identifies the number of available pages in the system execution space that are backed with real storage frames below 2 GB.

**Backed>2G**

identifies the number of available pages in the system execution space that are backed with real storage frames above 2 GB.

**Unbacked**

identifies the number of available pages in the system execution space that are not backed.

SXS pages in use:

**Total**

identifies the total number of pages in the system execution space that are currently in use:

```
Total = Alias + FreeStorage + Nucleus/Prefix + Other
```

**Alias**

identifies the number of pages in the system execution space that have an alias.

**FreeStorage**

identifies the number of pages in the system execution space that are being used for free storage requests.

**Nucleus/Prefix**

identifies the number of pages in the system execution space that are being used for the CP nucleus and the prefix pages.

**Other**

identifies the number of pages in the system execution space that are being used for purposes other than those listed elsewhere in the response.

Free Storage:

**Backed<2G**

identifies the number of pages being used for free storage requests that are backed with real storage frames below 2 GB.

## QUERY SXSPAGES

### **Backed>2G**

identifies the number of pages being used for free storage requests that are backed with real storage frames above 2 GB.

### **ReservedPages**

identifies the number of free storage reserved pages that are available for use.

Locked SXS pages:

### **Total**

identifies the total number of pages in the system execution space that are locked.

### **ByCommand**

identifies the number of pages in the system execution space that are locked by a command.

Deferred SXS page requests:

### **Alias**

identifies the number of requests that are waiting for a page for an alias.

### **Backed<2G**

identifies the number of requests that are waiting for a page backed with a real storage frame below 2 GB.

### **BackedAny**

identifies the number of requests that are waiting for a page backed with a real storage frame either above or below 2 GB.

### **Frame<2G**

identifies the number of requests that have a page but are waiting for a backing frame in real storage below 2 GB.

### **FrameAny**

identifies the number of requests that have a page but are waiting for a backing frame in real storage either above or below 2 GB.

### **Total**

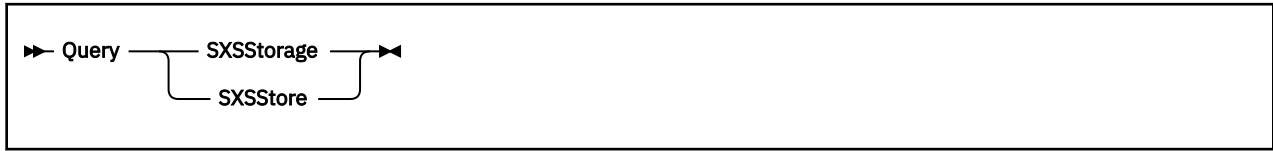
identifies the total number of system execution space page requests that are waiting to be satisfied:

```
Total = Alias + Backed<2G + BackedAny + Frame<2G + FrameAny
```

## **Messages**

- HCP263E Too many operands were supplied for this command.

# QUERY SXSSTORAGE



## Authorization

Privilege Class: A, B, E

## Purpose

Use QUERY SXSSTORAGE to display the host logical storage address ranges that are in use or usable by the system for the system execution space (SXS). The command response provides the following information:

- System execution area: this area contains the CP nucleus, the prefix pages, dynamically allocated CP-use pages (including free storage pages), aliases (references to guest pages), and routines loaded with the CPXLOAD system configuration statement or the CP CPXLOAD command.

The system execution area resides below 2 GB in the SXS.

- Frame table: there is one entry in this table for every real frame in the logical partition, up to the maximum amount of real storage supported by z/VM. Each frame table entry identifies the state of the frame. For example, the frame is online and therefore addressable by z/VM, or the frame is offline and not in use. The SET STORAGE command can be used to bring frames online or to take frames offline.

The frame table resides above 2 GB in the SXS.

- System execution space page management table: there is one entry in this table for every page within the system execution area to identify its usage, including whether it is backed by a real frame, and if that frame is below 2 GB or above 2 GB.

The system execution space page management table resides above 2 GB in the SXS.

- Frame table active range: this section of the frame table bounds the online portion of real storage. The lower value corresponds to absolute frame zero, and the upper value identifies the highest addressable frame. Some of the frames within this range might be offline.

To display the status of pages in the system execution area, use the QUERY SXSPAGES command.

## Responses

Response 1:

```
System Execution Area      nnnnnnnnnnnnnnnnn - nnnnnnnnnnnnnnnnn
Frame Table                nnnnnnnnnnnnnnnnn - nnnnnnnnnnnnnnnnn
SXS Page Management Table  nnnnnnnnnnnnnnnnn - nnnnnnnnnnnnnnnnn
Frame Table Active Range   nnnnnnnnnnnnnnnnn - nnnnnnnnnnnnnnnnn
```

**Note:** If SET UNDERSCORE ON is in effect, an underscore character ( \_ ) is used to divide each 16-digit address into two 8-digit entities.

## Messages

- HCP263E Too many operands were supplied for this command.

## QUERY SYSASCII

---

► Query — SYSAscii ◄

### Authorization

Privilege Class: B

### Purpose

Use QUERY SYSASCII to determine the status of the ASCII console.

### Usage Notes

1. A VIRTUAL version of this command, QUERY VIRTUAL SYSASCII, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL SYSASCII” on page 1337](#).

### Responses

Response 1:

```
SYSASCII free status
```

is the response to QUERY SYSASCII when the ASCII console is available and not attached to a guest.

#### **status**

is either *active* or *inactive*, corresponding to whether the ASCII console session on the Hardware Management console (HMC) is open or not open.

Response 2:

```
SYSASCII attached to userid status
```

is the response to QUERY SYSASCII when the ASCII console is available and attached to a guest.

#### **userid**

is the user ID of the specified guest that has the ASCII console attached.

#### **status**

is either *active* or *inactive*, corresponding to whether the ASCII console session on the Hardware Management Console (HMC) is open or not open.

### Messages

- HCP046E SYSASCII offline
- HCP263E Too many operands were supplied for this command.



## QUERY SYSCONTROL

---

► Query — SYSCONTROL ◄

### Authorization

Privilege Class: A, E

### Purpose

Use QUERY SYSCONTROL command to determine the current values of the algorithm settings controlled by the SET SYSCONTROL command. These values should be changed only when advised by IBM support personnel.

## QUERY SYSOPER

---

► Query — SYSOPER ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY SYSOPER to display the user ID of the current primary system operator.

### Usage Notes

1. Use the SET SYSOPER command to specify the user ID for the current primary system operator. For more information, see [“SET SYSOPER” on page 1731](#).
2. When you send messages to the user ID defined as the system operator, you can use the minimum abbreviation for OPERATOR, which is OP.

### Responses

Response 1:

When you enter a QUERY SYSOPER command, you receive the response:

```
Operator ID is userid.
```

Response 2:

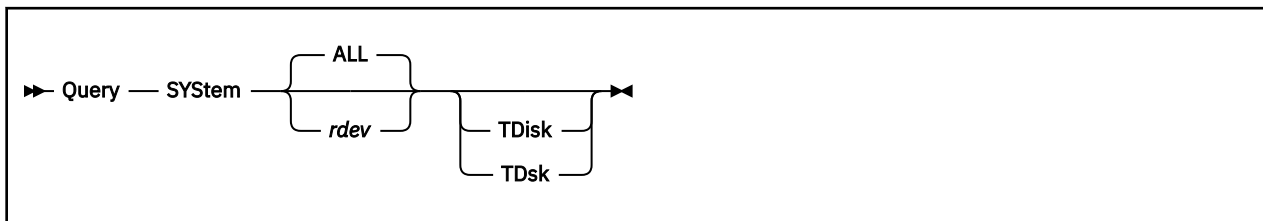
If no operator ID is currently defined, the response is:

```
No operator ID is currently active.
```

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*

# QUERY SYSTEM



## Authorization

Privilege Class: B

## Purpose

Use QUERY SYSTEM to display current user access to a system DASD volume.

## Operands

### *rdev*

is the real device number of a system DASD volume. Specify *rdev* when you want information about a single system device.

### ALL

requests a display of information about all system volumes. This is the default.

### TDisk

### TDsk

requests a display of information only about temporary disk (TDISK) usage on the volume or volumes.

## Usage Notes

1. You can use QUERY SYSTEM to display usage information about all user minidisks or only temporary disks on system DASD volumes. If you do not specify a choice, CP displays information about all user minidisks, including all temporary disks, on the volumes.
2. In an SSI cluster where there is sharing of real DASD among multiple z/VM images, QUERY SYSTEM will display only the user IDs of users on the system from which the query is made. It will not display user IDs of users linked on other systems in the SSI cluster.

## Responses

Response 1:

```

DASD rdev ATTACHED {SYSTEM} nnnn {volid}
                   {CPVOL }   {HYPERPAV ALIAS}
  
```

### *rdev*

identifies the real device number of the DASD volume.

### SYSTEM

indicates that the volume is attached to the system.

### CPVOL

indicates that the volume is owned by CP.

### *nnnn*

identifies the number of active minidisks on the volume or the number of users sharing the HyperPAV alias.

**valid**

identifies the volume label.

**HYPERPAV ALIAS**

indicates that the device is a system attached HyperPAV alias.

Response 2:

```
userid vdev xxxx[,... ] (up to three responses per line)
```

**userid**

identifies the users who have links to minidisks on the real device or users who share access to the real HyperPAV alias device.

**vdev**

identifies the virtual address by which the user refers to the minidisk or HyperPAV alias.

**xxxx**

identifies the type of access the user has, either R/O or R/W, for a minidisk, R/W for a HyperPAV alias, or the number of cylinders (CKD/ECKD) or blocks (FBA) of temporary disk space allocated. For some volumes such as CP-owned, the type of access is suppressed.

Response 3:

```
DASD MDISKS NOT FOUND
```

indicates that no active minidisks have been defined on this DASD volume.

Response 4:

```
DASD TDSKS NOT FOUND
```

indicates that no temporary disk space was defined on the indicated volume.

Response 5:

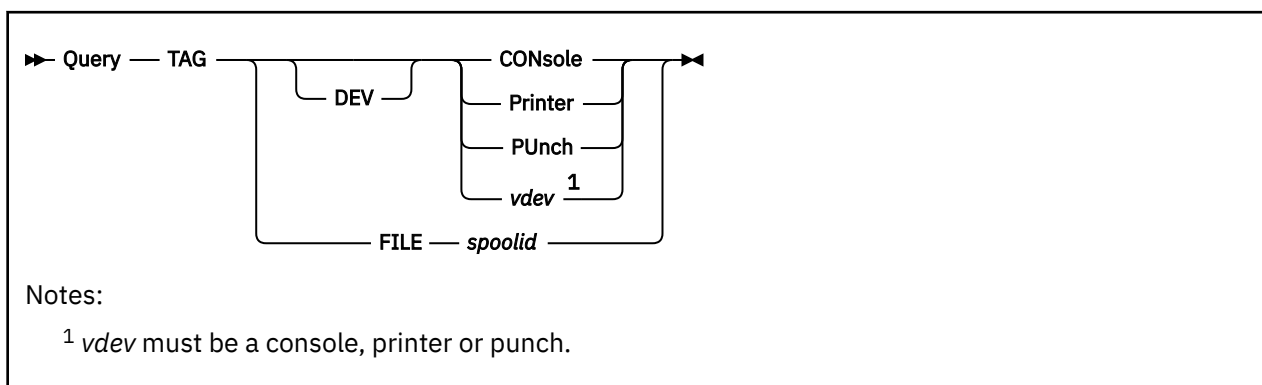
```
NO VIRTUAL HYPERPAV ALIASES DEFINED
```

indicates that no guests have defined HyperPAV aliases to share access to the real system attached HyperPAV alias device.

### Messages

- HCP002E Invalid operand - *operand*
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP1008E Device *rdev* is not a DASD
- HCP1009E Device *rdev* is not attached to the system

## QUERY TAG



### Authorization

Privilege Class: G

### Purpose

Use QUERY TAG to display the tag information associated with a specific virtual spool device or file.

### Operands

#### DEV

is an optional keyword which VM/ESA accepts for compatibility reasons.

#### CONsole

#### Printer

#### PUnch

#### *vdev*

specifies the virtual spooling device whose tag information you want to display. You can specify the virtual device number *vdev* of a device if you want to query the tag information associated with a virtual spooling device at a particular device location.

#### FILE *spoolid*

specifies that you want to display the tag information associated with a previously closed spool file owned by your virtual machine. The operand *spoolid* is the system-assigned spool file identifier that belongs to the file.

### Usage Notes

1. You can control the format and content of the tag field using the TAG command.
2. If an external security manager is installed on your system, you may not be authorized to enter this command. For additional information, contact your security administrator.

### Responses

Response 1:

```
type vdev TAG:
tagtext
```

displays the tag information associated with a virtual spooling device.

#### *type*

identifies the type as PRT, PUN, or CONS.

## QUERY TAG

### ***vdev***

identifies the virtual device number.

### ***tagtext***

identifies the tag information associated with the device. This tag information may be up to 136 characters long.

Response 2:

```
tagtext
```

- or -

```
(TAG BLANK)
```

displays the tag information associated with a specified spool file.

### ***tagtext***

identifies the tag information associated with the file. This tag information may be up to 136 characters long.

### **(TAG BLANK)**

is displayed when the tag is all blanks.

Response 3:

```
type {vdev } TAG NOT SET
```

is the response if:

- You have not specified tag information for a device
- You have explicitly set the tag field to blanks.

### ***type***

identifies the type as PRT, PUN, or CONS.

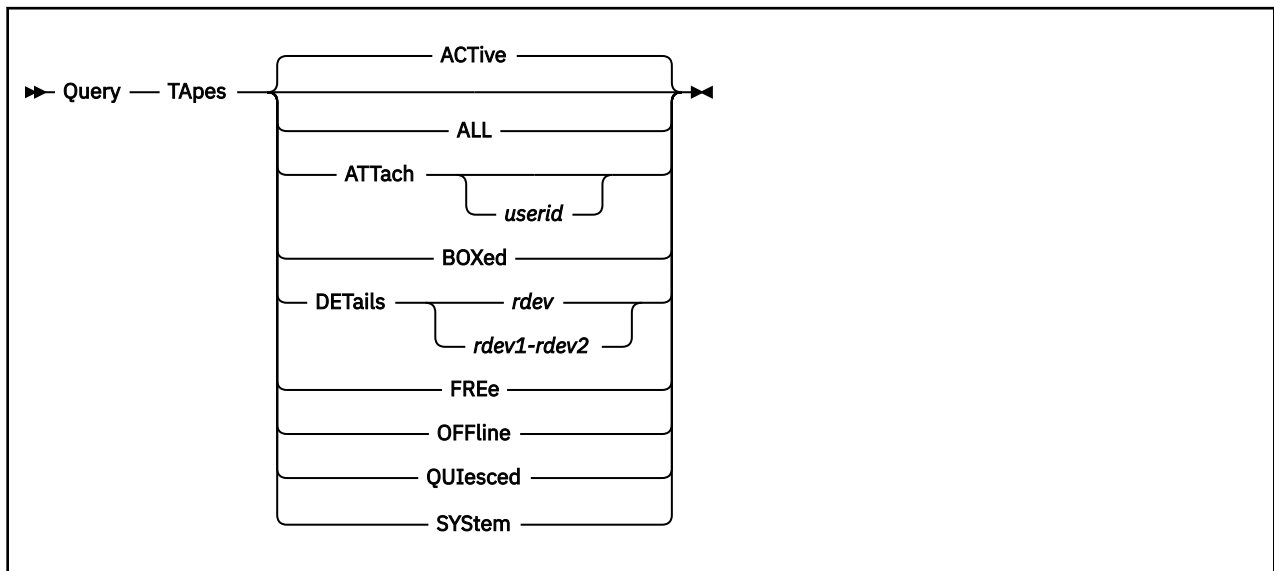
### ***vdev***

identifies the virtual device number.

## **Messages**

- HCP002E Invalid operand - *operand*
- HCP016E TAG invalid for specified file type
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP035E Device type missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP413E Command processing cannot complete due to a spool paging error.

## QUERY TAPES



### Authorization

Privilege Class: B

### Purpose

Use QUERY TAPES to display the status of real magnetic tape drives.

### Operands

#### ACTive

indicates active magnetic tape drives. An active device is one that is in use by a user or the system and is specified as neither FREE nor OFFLINE. This is the default.

#### ALL

indicates all magnetic tape drives attached to the system. The status is displayed in the order of ACTIVE, FREE, and OFFLINE. You receive the same response as you would receive if you entered all of the following commands:

```
query tapes active
query tapes free
query tapes offline
```

#### ATTach *userid*

indicates the magnetic tape drives that are attached to a user on the system. If you specify *userid*, only tape drives attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all tape drives attached to any user on the system. Attached drives are always active.

#### BOXed

indicates the magnetic tape drives that are unavailable (usually because of a hardware condition). A boxed drive may also be active, free, or offline.

#### DETailS

displays detailed information about real tapes.

## QUERY TAPES

### **rdev**

#### **rdev1-rdev2**

is the real device number (*rdev*), a range of real device numbers (*rdev1-rdev2*), or some combination of those items.

For tapes attached to a library, the information provided is:

- Library Identifier (sequence number)
- Logical Device Group Identifier (libport)

For tapes not attached to a library, the information is not relevant and is returned as a string of dashes ("-").

For tapes that are capable of hardware encryption, this information will also be presented, along with any encryption settings that were established via the ATTACH or SET RDEV commands.

### **FREE**

indicates all free magnetic tape drives. A free drive is one that is not currently in use by a user or the system and is specified as neither ACTIVE nor OFFLINE.

### **OFFline**

indicates all offline magnetic tape drives. An offline drive is one that is specified as neither ACTIVE nor FREE. This offline response will also show offline drives that are quiesced.

### **QUIesced**

indicates all quiesced magnetic tape drives. An quiesced drive is an OFFLINE drive that cannot be varied online until a resume request is received for the drive.

### **SYStem**

indicates the drives in use by the system. A system device is always active.

## Usage Notes

1. A VIRTUAL version of this command, QUERY VIRTUAL TAPES, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL TAPES”](#) on page 1338.

## Responses

If the system is using a magnetic tape drive that is in an intervention-required state and you enter QUERY TAPES, you receive a response like this:

```
Tape 0104 has intervention required.
```

0104 is the real device number of the tape drive that needs the operator's intervention.

**Note:** This response will be preceded by one of the following.

Response 1:

If you enter a QUERY TAPES, QUERY TAPES ACTIVE, or QUERY TAPES ATTACH command and there is a tape drive dedicated to a user, you see something like this:

```
q tapes attach
TAPE 0102 ATTACHED TO RALPH 0236 R/W
Ready;
```

This means the magnetic tape drive at the address 0102 is attached to RALPH's virtual machine as the virtual device number 0236. R/W means the user RALPH has read-write access to the tape drive. R/W might also read R/O, which means the user has read-only access.

Response 2:

If the tape drive is currently being used by the TRSAVE command, you see something like this:



```
q tapes
TAPE 0242 ASSIGNED SYSTEM TRSAVE
Ready;
```

## Response 3:

If a tape that is not dedicated to a user is not available (boxed) or you enter a QUERY TAPES BOXED command, you see something like this:

```
q tapes boxed
TAPE 0244 BOXED
Ready;
```

## Response 4:

If you enter a QUERY TAPES ALL command, CP displays the same response as it would display if you entered QUERY TAPES ACTIVE, QUERY TAPES FREE, and QUERY TAPES OFFLINE commands.

## Response 5:

If CP cannot find a tape drive of the type you specified, you see a response like this:

```
q tapes offline
An offline tape was not found.
Ready;
```

In this response, *offline* may be replaced with *active*, *attached*, *BOXED*, *free*, depending on what you specified on the command.

## Response 6:

If you enter a QUERY TAPES, QUERY TAPES ACTIVE, QUERY TAPES ALL, or QUERY TAPES ATTACH command, and the tape drive was previously attached with the MULTIUSER option of the CP ATTACH command or the MULTIUSER operand of the DEDICATE directory statement, you see a response like this:

```
q tapes attach
TAPE 2100 ATTACHED TO USER1    0181 R/W  NOASSIGN MULTIUSER
TAPE 2100 ATTACHED TO USER2    0181 R/W  NOASSIGN MULTIUSER
Ready;
```

This means that the magnetic tape drive at real address 2100 is attached to USER1 and USER2 at virtual device address 0181. MULTIUSER indicates that the tape drive is shared by both users, and NOASSIGN states that the users must manage the assignment of the tape drive by using *assign* and *unassign* I/O.

## Response 7:

If you enter a QUERY TAPES DETAILS *rdev(s)* command, and the tape drive is not a part of the library, you will see a response like this:

```
q tapes details 4499
16:06:22 TAPE 4499 SEQUENCE NUMBER ----- LIBPORT -
Ready; T=0.01/0.01 16:06:22
```

For a device that is part of a library, you will see a response like this:

```
q tapes details 4300
20:04:04 TAPE 4300 SEQUENCE NUMBER 00103 LIBPORT 1
Ready; T=0.01/0.01 20:04:04
```

**Note:** For the 3592 Model E07 and higher, an additional line showing the current tape block position and relative position will be shown. See the next response below for an example.

## Response 8:

If you enter a query TAPES DETAILS command, and the tape drive is capable of hardware encryption, you will see a response like this:

```
q tapes details BB00
TAPE BB00 SEQUENCE NUMBER E0010 LIBPORT 1 ENCRYPTION CAPABLE
```

## QUERY TAPES

```
ACTIVE KEY LABEL(S):  
  (L) existing key label one  
  (H) a different key label used with a hash mechanism  
ATTACHED KEY LABEL(S):  
  (L) the first mighty key label  
  (H) the second mighty key label  
INACTIVE KEY LABEL(S): DEFAULT  
Position: 1350147 BLK Relative position: 005%  
Ready; T=0.01/0.01 00:11:09
```

### Note:

1. The ATTACHED KEY LABEL(S) are only displayed if they differ from what is currently being used on the tape cartridge, such as when data is being appended to an existing tape.
2. An encoding mechanism of "label" means the key label is used directly, while a "hash" encoding mechanism creates a hash out of the key label and provides more flexibility in sharing encrypted data.
3. The "inactive" keys are specified on a SET RDEV command, while the "active" keys are either specified on an ATTACH command, or are promoted from the "inactive" setting when an ATTACH does not include encryption keys.
4. The last line, showing the current tape block position and relative position, is shown only for the 3592 Model E07 and higher.

### Messages

- HCP003E Invalid option - *option*
- HCP009E An invalid address range was specified.
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device *rdev* does not exist
- HCP263E Too many operands were supplied for this command.

## QUERY TDISK



### Authorization

Privilege Class: B

### Purpose

Use QUERY TDISK to display all defined and assigned temporary disk (TDISK) space.

### Responses

Response 1:

```
DASD rdev ATTACHED CPVOL nnnn valid
```

#### ***rdev***

identifies the real device number of the DASD volume.

#### **CPVOL**

indicates that the volume is a CP volume.

#### ***nnnn***

identifies the number of temporary disk links on this volume.

#### ***valid***

identifies the volume label.

Response 2:

```
userid vdev nnnn [,...] (up to 3 responses per line)
```

#### ***userid***

identifies the users who have temporary disk links on *rdev*.

#### ***vdev***

identifies the virtual address by which the user refers to the tdisk.

#### ***nnnn***

identifies the number of cylinders (CKD/ECKD) or blocks (FBA) of temporary disk space allocated.

Response 3:

```
DASD TDSKS NOT FOUND
```

indicates that no active temporary DASD have been defined.

### Messages

- HCP002E Invalid operand - *operand*

## QUERY TDISKCLR

---

► Query — TDISKCLR ◄

### Authorization

Privilege Class: B

### Purpose

Use QUERY TDISKCLR to display the status of the TDISK clearing function. This status depends on the setting in the FEATURES system configuration statement.

### Responses

Response 1:

```
Temporary Disk Security is active.
```

is displayed if all TDISK space is currently being cleared whenever a DASD volume is attached to the system or whenever a TDISK minidisk is being detached from a user.

Response 2:

```
Temporary Disk Security is inactive.
```

is displayed if no TDISK clearing is being performed.

### Messages

- HCP263E Too many operands were supplied for this command.

# QUERY TERMINAL

► Query — TERminal ◄

## Authorization

Privilege Class: G

## Purpose

Use QUERY TERMINAL to display the options active for your virtual console.

## Usage Notes

1. The SCRNSAVE function of the TERMINAL command is not supported by z/VM. It is included with its default setting only for compatibility with VM/SP HPO.
2. CP displays the TYPE, PROMPT, SCROLL, CNTL, and ASCIITBL messages only when you enter the command from a TTY device.

## Responses

Response 1:

```

LINEND {char}, LINEDEL {char}, CHARDEL {char}, ESCAPE {char}, TABCHAR {char}
      {OFF} {OFF} {OFF} {OFF} {OFF}

LINESIZE {nnn}, ATTN {ON}, APL {ON}, TEXT {ON}, MODE {CP}, HIGHLIGHT {ON}
      {OFF} {OFF} {OFF} {OFF} {VM} {OFF}

CONMODE {3215}, BREAKIN {IMMED}, BRKKEY {PA1}, SCRNSAVE {OFF}
      {3270} {GUESTCTL} {PA2}
      {NONE} {PFnn}
      {CLEAR}
      {NONE}

TYPE {3101}, PROMPT {VM}, SCROLL {nnn}, CNTL {SYS}, ASCIITBL {VM1}
      {TTY} {TTY} {CONT} {USR} {VM2}

AUTOOCR {ON}, MORE nnn mmm, HOLD {ON}, TIMESTAMP {CP}, SYS3270 {ON}
      {OFF} {OFF} {ON} {OFF} {OFF}
      {OFF}

PRECEDENCE {SECUSER}
      {STARMSG}

```

### LINEND char

#### LINEND OFF

displays the logical line end symbol in effect for your virtual machine (*char*), or indicates that no logical line end symbol is allowed (OFF).

### LINEDEL char

#### LINEDEL OFF

displays the logical line delete symbol in effect for your virtual machine (*char*), or indicates that no logical line delete symbol is allowed (OFF).

### CHARDEL char

#### CHARDEL OFF

displays the logical character delete symbol in effect for your virtual machine (*char*), or indicates that no logical character delete symbol is allowed (OFF).

**ESCAPE char****ESCAPE OFF**

displays the logical escape character in effect for your virtual machine (*char*), or indicates that no logical escape character is allowed (OFF).

**TABCHAR char****TABCHAR OFF**

displays the tab character symbol in effect for your virtual machine (*char*), or indicates that no logical tab character is defined (OFF).

**LINESIZE nnn****LINESIZE OFF**

specifies the maximum allowable line length for terminal output that is in effect for your virtual machine. OFF specifies that the terminal output line length is not adjusted by CP.

**ATTN ON****ATTN OFF**

indicates whether the attention signal is to be reflected on keyboard or printer terminals. If ON, the printer reflects it as an exclamation point (!) followed by a carriage return.

**APL ON****APL OFF**

indicates whether CP is to use the APL terminal translation tables or the normal translation tables. If APL is ON, CP uses the APL translation tables.

**TEXT ON****TEXT OFF**

indicates whether CP is to use the TEXT terminal translation tables or the normal terminal translation tables. If TEXT is ON, CP uses the TEXT translation tables.

**MODE CP****MODE OFF**

indicates whether one or more attentions force your virtual machine into the CP environment, or whether one or more attentions are reflected to your virtual machine. If MODE is CP, one attention forces your virtual machine into the CP environment.

**HILIGHT ON****HILIGHT OFF**

ON means CP highlights the command redisplay line. OFF means CP displays the command redisplay line at normal intensity.

**CONMODE 3215****CONMODE 3270****CONMODE NONE**

indicates the console mode of your virtual console. 3215 indicates that CP provides 3215 CCW processing, and 3270 indicates that it provides 3270 CCW processing. NONE indicates that your virtual machine does not currently have a console defined.

**BREAKIN IMMED****BREAKIN GUESTCTL**

lets you control break-in by CP for full-screen guests. The GUESTCTL option specifies that, while in full-screen mode, messages from CP are displayed only when you press the BRKKEY key. If you specify GUESTCTL, CP break-in still occurs if you request a CP function or if a priority message is pending for your virtual machine. GUESTCTL is specified using TERMINAL BREAKIN. If you specify IMMED, which is the default, break-in by CP takes place on any full-screen READ and full-screen WRITE that has an inactive keyboard. Break-in also takes place when you press the BRKKEY.

**BRKKEY PA1****BRKKEY PA2****BRKKEY PFnn****BRKKEY CLEAR****BRKKEY NONE**

displays the key that controls break-in by CP for full-screen guests (to allow you to return to CP mode). NONE indicates that there is no BRKKEY defined or it is disabled.

**SCRNSAVE OFF**

is included with its default setting for reasons of compatibility.

**TYPE 3101****TYPE TTY**

3101 identifies the terminal as a display device, and 3101 functions are enabled. TTY identifies the terminal as a typewriter terminal, and 3101 functions are not in effect.

**PROMPT VM****PROMPT TTY**

VM indicates that VM enables the line feed prompting sequence. TTY indicates that VM enables the normal TTY prompting sequence of a period in column 1 and the cursor in column 2.

**SCROLL nnn****SCROLL CONT**

The operand *nnn* specifies the number of lines displayed before the status **\*\*\* MORE \*\*\*** is displayed. CONT specifies continuous scrolling without interruption.

**CNTL SYS****CNTL USR**

SYS specifies that CP inserts TTY line control characters for output directed to a TTY terminal. USR specifies that the application program inserts TTY line control characters for output directed to a TTY terminal.

**ASCIITBL VM1****ASCIITBL VM2**

VM1 specifies the use of the translation table based on the TTY ANSI X3.4 1977 standard. VM2 specifies the use of the translation table based on the TTY ANSI X3.26 1980 standard.

**AUTO CR ON****AUTO CR OFF**

ON specifies that a carriage return is performed following the execution of the 3215 CCW X'01' (write without carriage return to a graphic display operating in line mode). OFF specifies that a carriage return not be performed.

**MORE nnn mmm**

specifies the number of seconds that will elapse between system actions. *nnn* is the number of seconds between when the time when a MORE . . . is displayed and the terminal alarm sounds. *mmm* is the number of seconds between the time when the alarm sounds and the screen is cleared.

**HOLD ON****HOLD OFF**

ON means CP displays a status of HOLDING when the screen fills up and at least one highlighted message appears on the screen. OFF means CP does not display a status of HOLDING. You can still cause CP to display a HOLDING status by pressing the Enter key when CP has displayed a MORE . . . status.

**TIMESTAMP CP****TIMESTAMP ON****TIMESTAMP OFF**

CP means control program messages and command responses on the console are prefixed with the local time of day. OFF means messages are displayed with no prefix. ON causes all virtual machine and CP messages to be identified with time stamps.

**SYS3270 ON****SYS3270 OFF**

ON means the virtual console emulates an integrated 3270 device. OFF means the virtual console does not emulate an integrated 3270 device.

**PRECEDENCE SECUSER****PRECEDENCE STARMMSG**

The PRECEDENCE setting determines the order of preference for delivering virtual console output that is eligible for routing both to a secondary user and through the primary user's connection to the \*MSG or \*MSGALL IUCV system service. SECUSER means that such output should be sent to a secondary

## QUERY TERMINAL

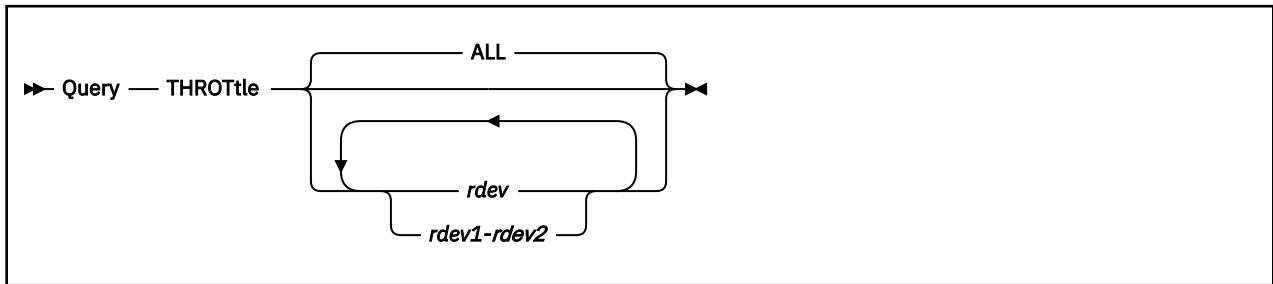
user. STARMMSG means that such output should be sent through IUCV. See [“TERMINAL PRECEDENCE”](#) on page 1938 for details.

### Messages

- HCP263E Too many operands were supplied for this command.



## QUERY THROTTLE



### Authorization

Privilege Class: B, E

### Purpose

Use QUERY THROTTLE to display the devices being throttled and their I/O rates.

### Operands

#### ALL

tells CP to display all the real devices that are being throttled, their I/O rates and any delays that have occurred for the given devices. ALL is the default.

#### *rdev*

#### *rdev1-rdev2*

is the real device number or range of real device numbers for which you want I/O throttling information.

### Usage Notes

1. To change I/O throttling on your system, use the SET THROTTLE command (see [“SET THROTTLE”](#) on page 1734).

### Responses

Response 1:

To display all of the real devices in the I/O throttling list and their current I/O rates, enter the following QUERY THROTTLE command:

```
query throttle all
I/O throttling is currently active. 2 device(s) are in the list
Device 0701 has 6 delays. Rate is 1 I/Os per second
Device 0702 has 0 delays. Rate is 2 I/Os per second
Ready;
```

Response 2:

To query throttling of a specified device enter the following QUERY THROTTLE command:

```
query throttle 702
Device 0702 has 0 delays. Rate is 2 I/Os per second
Ready;
```

Response 3:

To query throttling for a range of devices enter the following QUERY THROTTLE command:

## QUERY THROTTLE

```
query throttle 701-704
Device 0701 has 6 delays. Rate is 1 I/Os per second
Device 0702 has 0 delays. Rate is 2 I/Os per second
Device 0703 has 0 delays. Rate is 4 I/Os per second
Device 0704 is not in the I/O throttling list
Ready;
```

Response 4:

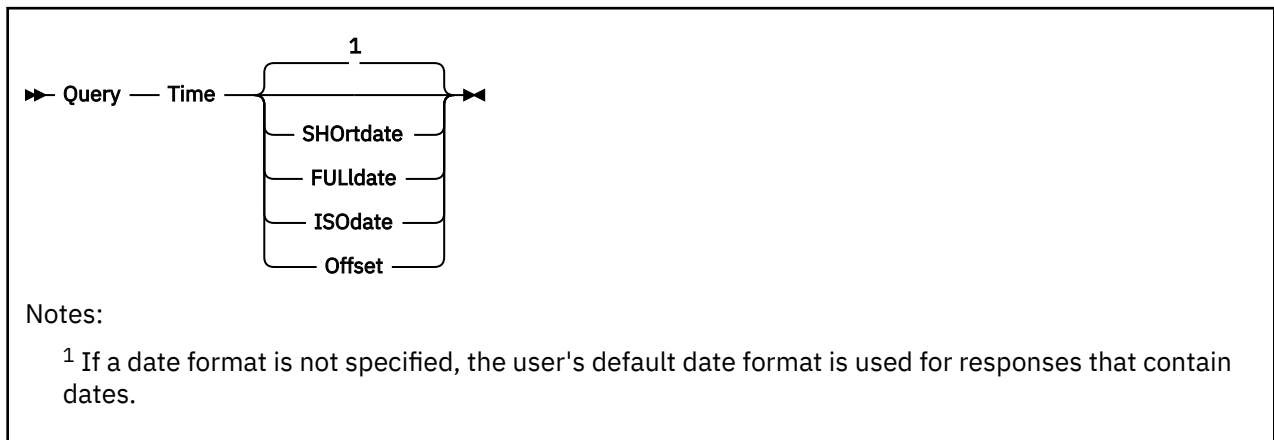
If no throttling is active:

```
query throttle
I/O throttling is not currently active
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP009E An invalid address range was specified.
- HCP040E Device *rdev* does not exist
- HCP1150E DASD *rdev* is not a valid base exposure.

## QUERY TIME



### Authorization

Privilege Class: G

### Purpose

Use QUERY TIME to display the following:

- Current time
- Time zone
- Day of the week
- Date
- Connect time for the current session
- Processor time for the current session

### Operands

#### SHOrtdate

specifies that dates in the response to this command will be displayed in *mm/dd/yy* format, where *mm* is the month, *dd* is the day of the month, and *yy* is the 2-digit year.

#### FULldate

specifies that dates in the response to this command will be displayed in *mm/dd/yyyy* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

#### ISOdate

specifies that dates in the response to this command will be displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

#### Offset

indicates that you want to see the difference between the system's current time zone and Coordinated Universal Time (UTC).

### Usage Notes

1. If the wait-state interpretation capability is available on the real processor, it is used for dedicated virtual CPUs. When the wait-state interpretation capability is used, the guest virtual machine remains dispatched even when it enters a PSW enabled-wait state. Time consumed dispatched in the wait state is included in the virtual and total CPU time reported for the guest.

## QUERY TIME

2. The default date format for certain CP and CMS commands can be set on a system-wide basis and also for the individual user. The system-wide default date format is set with the `SYSTEM_DATEFORMAT` system configuration statement. The user's default date format is set with the `DATEFORMAT` directory statement. The system-wide default and the user's default can also be set with the `CP SET DATEFORMAT` command. The user's default date format defaults to the system-wide default. The system-wide and user settings can be queried with the `CP QUERY DATEFORMAT` command. The hierarchy of possible date format settings for the `QUERY TIME` command, from highest priority to lowest, is:

- `QUERY TIME` command operand
- User default
- System-wide default

### Responses

Response 1:

If you enter `QUERY TIME SHORTDATE`, you get the following response:

```
TIME IS hh:mm:mm zone weekday mm/dd/yy  
CONNECT= hh:mm:ss VIRTCPU= mmm:ss.hs TOTCPU= mmm:ss.hs
```

This shows the current real clock time (in hours, minutes, and seconds), the time zone (for example, EST), the day of the week, and the calendar date.

Response 2:

If you enter `QUERY TIME FULLDATE`, you get the following response:

```
TIME IS hh:mm:mm zone weekday mm/dd/yyyy
```

```
CONNECT= hh:mm:ss VIRTCPU= mmm:ss.hs TOTCPU= mmm:ss.hs
```

Response 3:

If you enter `QUERY TIME ISODATE`, you get the following response:

```
TIME IS hh:mm:mm zone weekday yyyy-mm-dd
```

```
CONNECT= hh:mm:ss VIRTCPU= mmm:ss.hs TOTCPU= mmm:ss.hs
```

*Explanation of Response 1, 2, and 3 Contents:*

#### **CONNECT= hh:mm:ss**

identifies the actual clock time consumed in the current session in hours, minutes, and seconds. The maximum value for this field is 99:59.59. A time value larger than the maximum will be indicated by this value.

#### **VIRTCPU= mmm:ss.hs**

identifies the virtual processor time consumed in the current session in minutes, seconds, and hundredths of seconds. The maximum value for this field is 999:59.99. A time value larger than the maximum will be indicated by this value. If multithreading is enabled, this is reported as MT-1 equivalent time.

#### **TOTCPU= mmm:ss.hs**

identifies the total processor time (virtual and overhead) consumed in the current session in minutes, seconds, and hundredths of seconds. The maximum value for this field is 999:59.99. A time value larger than the maximum will be indicated by this value. If multithreading is enabled, this is reported as MT-1 equivalent time.

Response 4:

If you enter `QUERY TIME OFFSET`, you get the following response:

```
Time zone zone is hh.mm.ss direction of UTC.
```

shows the current time zone's offset from UTC, Coordinated Universal Time.

***zone***

is the identifier (for example, EST) that indicates the current time zone.

***hh.mm.ss***

is the number of hours, minutes, and seconds that the current time is offset from UTC.

***direction***

is the geographical direction of this time zone from longitude 0. *direction* may be EAST or WEST.

You could also receive this response:

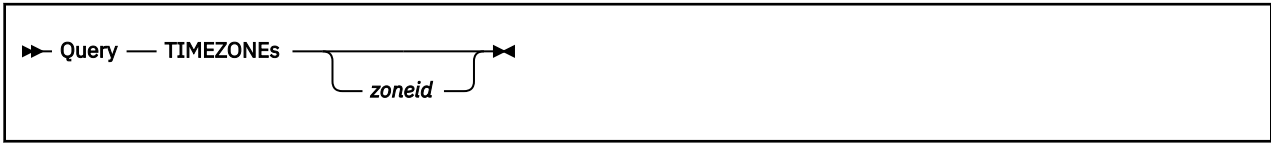
```
Time zone zone is equivalent to UTC.
```

This means the value for the adjustment from UTC (the *hh.mm.ss* value) is zero.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

# QUERY TIMEZONES



## Authorization

Privilege Class: G

## Purpose

Use QUERY TIMEZONES to display the list of active and inactive time zone definitions on your system.

## Operands

### *zoneid*

specifies the time zone ID definition that you wish to display. When *zoneid* is omitted, CP displays information about all defined time zones and identifies the active time zone.

## Usage Notes

1. To define new time zones or to change existing time zone characteristics, use the DEFINE TIMEZONE command. For more information, see [“DEFINE TIMEZONE” on page 225](#).
2. To select a new active time zone, use the SET TIMEZONE command. For more information, see [“SET TIMEZONE” on page 1738](#).
3. You can also define time zones and set the active time zone in the system configuration file by using the [TIMEZONE\\_DEFINITION Statement](#) and [TIMEZONE\\_BOUNDARY Statement](#). For more information, see *z/VM: CP Planning and Administration*.

## Responses

Response 1:

The QUERY TIMEZONES command provides this type of response:

Zone	Direction	Offset	Status
GMT	----	00.00.00	Inactive
UTC	----	00.00.00	Inactive
EDT	West	04.00.00	Active
EST	West	05.00.00	Inactive
PDT	West	07.00.00	Inactive
PST	West	08.00.00	Inactive
CET	East	01.00.00	Inactive
CDT	West	05.00.00	Inactive

The first column shows the time zone ID (*zoneid*). This identifier can be used in a SET TIMEZONE command. CP automatically defines the Greenwich Mean Time (GMT) and Coordinated Universal Time (UTC) time zones.

The second column shows the geographical direction of this time zone from longitude 0. If you define a time zone with an offset of 00.00.00, CP does not display a direction.

The third column shows the offset (expressed in hours, minutes, and seconds) between the time zone and the UTC time zone.

The fourth column tells whether the time zone is currently active. Only one time zone can be active at one time.

Response 2:

When in an STP environment that derives time zone information from the STP server, the QUERY TIMEZONES command lists only the time zones provided by STP as well as the default GMT/UTC. The format of the boundary output is dependent on the issuing user's dateformat setting. The response looks like the following:

Zone	Direction	Offset	Status	Boundary
GMT	----	00.00.00	Inactive	
UTC	----	00.00.00	Inactive	
EDT	West	04.00.00	Active-(STP)	
EST	West	05.00.00	Inactive-(STP)	10/31/09 02:00:00

If the Adjust Time Zone panel on the HMC is configured to MANUAL, and no boundary has been configured then the boundary output defaults to 'Not Scheduled' as shown in the following:

Zone	Direction	Offset	Status	Boundary
GMT	----	00.00.00	Inactive	
UTC	----	00.00.00	Inactive	
EDT	West	04.00.00	Active-(STP)	
EST	West	05.00.00	Inactive-(STP)	Not Scheduled

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP6706E Invalid time zone - *zoneid*

## QUERY TOKEN



### Authorization

Privilege Class: B

### Purpose

Use QUERY TOKEN to display the current or target configuration token for your machine. CP uses this token and the channel subsystem when making dynamic I/O changes.

### Operands

#### CURrent

tells CP to display the current configuration token.

#### TARget

tells CP to display the target configuration token.

### Usage Notes

1. The *current* configuration token is a string value associated with the present I/O configuration.
2. The *target* configuration token is a string value associated with the I/O configuration after a dynamic I/O configuration change has completed successfully.
3. For more information on configuration tokens see the [SAY Statement](#) in *z/VM: CP Planning and Administration*.

### Responses

Response 1:

To display the current token, enter the following:

```
query token current
The current VM configuration token is:
VM-TOKEN05/03/9507:02:51    ...05/03/9506:42:27
Ready;
```

If your system is not allowed to make dynamic I/O changes you will see the following:

```
query token current
HCPCC06816E Dynamic I/O changes are not allowed on this system
The current channel subsystem configuration token is:
PKSTYAC1 99-12-10 07:24:43 WRY    IODF
Ready(06816);
```

Response 2:

To display the target token, enter the following:

```
query token target
The target VM configuration token is:
VM-TOKEN05/03/9507:02:51    ...05/03/9506:42:27
Ready;
```



If your system is not allowed to make dynamic I/O changes you will see the following:

```
query token target
HCPCC06816E Dynamic I/O changes are not allowed on this system
The target channel subsystem configuration token is:
PKSTYAC1 99-12-10 07:24:43 WRY      IODF
Ready(06816);
```

## Messages

- HCP0002E Invalid operand - *operand*
- HCP0003E Invalid option - command contains extra option(s) starting with *option*
- HCP6706E Invalid string - *string*
- HCP6808E Cannot find the configuration token on the system
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Cannot find configuration token in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed on this system

## QUERY TRACE



### Authorization

Privilege Class: G

### Purpose

Use QUERY TRACE to display the TRACE settings.

### Operands

#### ACTive

identifies the default option and displays attributes of all traps within the active trace set. This is the default.

#### ALL

displays attributes of all traps within all trace sets.

#### *name*

displays attributes of all traps within the trace set specified by the 1- to 8-character name.

#### RETurns

displays TRACE CALL return path. No trap attributes are displayed.

#### SETS

displays all trace set names. No trap attributes are displayed.

### Usage Notes

1. QUERY PER is accepted as a synonym for QUERY TRACE.

### Responses

Response 1:

```
TRACE COUNT IS nnnnnnnn
```

precedes the display of trace traps if TRACE COUNT is active.

Response 2:

The following response is to QUERY TRACE SETS.

```
SETS: name1, name2, ...
```

#### *name1...*

identifies trace set names.

Response 3:

The following response is to QUERY TRACE RETURNS.

```
RETURNS: name1, name2,...
```

**name1...**

identifies trace sets in the trace return path.

Response 4:

```
No trace sets defined
```

is the response for any form of the QUERY TRACE command if no trace sets have been defined.

Response 5:

```
No trace calls active
```

is the response for QUERY TRACE RETURNS if no trace calls are active; therefore, there are no return paths.

Response 6:

The following responses are to QUERY TRACE ALL, QUERY TRACE ACTIVE, and QUERY TRACE *name*.

The current status of all defined trace sets, of trace traps within the active trace set, or of a specifically identified trace set, is displayed by QUERY TRACE. Each trace trap has a specific format as shown below where *ident* is the trace trap identifier for each trap within the trace set. A common header exists for these responses.

```
NAME setname (ACTIVE+[NO]TXSUSPEND)
              (ACTIVE+SUSPEND+[NO]TXSUSPEND)
              ([NO]TXSUSPEND)
```

**Notes:**

1. The parenthesized word ACTIVE appears in the "NAME *setname* ..." header line for the set that is currently active.
2. TXSUSPEND indicates that traps in this trace set will be suspended while the virtual CPU is in transactional-execution mode. See "[TRACE TXSUSPEND/NOTXSUSPEND](#)" on page 2003 for details.

Trace entries can be displayed as `STO raddr; ASN xxx` or as `ASCE raddr;`; the STO or ASCE information may be on a line by itself, depending on the length of the FROM and INTO information.

Query Trace Examples:

```
1  BRANCH FROM 0000000000000000-1111111111111111
    INTO 0000222222222222-0000555555555555
    TERM  NOPRINT NORUN SIM
    SKIP 00000 PASS 00000 STOP 00000 STEP 00000
    CMD  NONE

2  BRANCH FROM 0000000000000000-1111111111111111
    STO 00000000 ASN 0000
    INTO 0000222222222222-0000555555555555
    STO 00000000 ASN 0000
    TERM  NOPRINT NORUN SIM
    SKIP 00000 PASS 00000 STOP 00000 STEP 00000
    CMD  NONE

3  BRANCH FROM 00000000-00000011          STO 00000000 ASN 0000
    INTO 00002222-00005555          STO 00000000 ASN 0000
    TERM  NOPRINT NORUN SIM
    SKIP 00000 PASS 00000 STOP 00000 STEP 00000
    CMD  NONE

4  STORE FROM 00000000-00000011          STO 00000000 ASN 0000
    INTO 00000000000000222-0000055555555555
    STO 00000000 ASN 0000
    TERM  NOPRINT NORUN SIM
    SKIP 00000 PASS 00000 STOP 00000 STEP 00000
    CMD  NONE
```

Access Register Alteration Trap Format

```

ident  AREG  FROM  hexloc[-hexloc] {[STO raddr [ASN xxxx]]}
                                     {[ASCE raddr]}
DATA  range
REG(S)  n1  n2... MASK hexstring
{TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER]
{NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
SKIP n  PASS n  STOP n  STEP n
CMD...CP command...

```

Instruction Trap Format

```

ident  INSTR PSWA hexloc[-hexloc] {[STO sto [ASN xxxx]]}
                                     {[ASCE raddr]}
{TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER]
{NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
SKIP n  PASS n  STOP n  STEP n
CMD...CP command...

```

**Note:** *hexloc* (-*hexloc*) is the “range” specified on the trace instruction.

Store Trap Format

```

ident  STORE  FROM  hexloc[-hexloc] {[STO sto [ASN xxxx]]}
                                     {[ASCE raddr]}
INTO  hexloc[-hexloc] {[STO sto [ASNxxxx]]}
                       {[ASCE raddr]}
[ASIT asit] [SPACE owner:space_name]

[DATA hexstring]
[MASK hexstring]
{TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER]
{NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
SKIP n  PASS n  STOP n  STEP n
CMD...CP command...

```

Branch Trap Format

```

ident  BRANCH FROM  hexloc[-hexloc] {[STO sto [ASN xxxx]]}
                                     {[ASCE raddr]}
INTO  hexloc[-hexloc] {[STO sto [ASNxxxx]]}
                       {[ASCE raddr]}
{TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER]
{NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
SKIP n  PASS n  STOP n  STEP n
CMD...CP command...

```

General Register Alteration Trap Format

```

ident  GREG  FROM  hexloc[-hexloc] {[STO raddr [ASN xxxx]]}
                                     {[ASCE raddr]}
DATA  range
REG(S)  n1  n2... MASK hexstring
{TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER]
{NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
SKIP n  PASS n  STOP n  STEP n
CMD...CP command...

```

I/O Trap Format

```

ident  typename FROM  hexloc[-hexloc] {[STO raddr [ASN xxxx]]}
                                     {[ASCE raddr]}
HEX n[-m]
{TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER] [INST] [CCW]
{NOTERM} {NOPRINT} {NORUN} {NOSIM} [NODAT] [PROB] [INT ] [IODATA n]
SKIP n  PASS n  STOP n  STEP n
CMD...CP command...

```

Mnemonic Trap Format

```

ident  typename FROM  hexloc[-hexloc] {[STO raddr [ASN xxxx]]}
                                     {[ASCE raddr]}
{TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER]

```

```
{NOTERM} {NOPRINT} {NORUN}{NOSIM} [NODAT] [PROB ]
SKIP n PASS n STOP n STEP n
CMD...CP command...
```

### TRACE MCH Format

```
ident  typename FROM hexloc[-hexloc] {[STO raddr [ASN xxxx]]}
                                           {[ASCE raddr]}
{TERM } {PRINT } {RUN } {SIM }
{NOTERM} {NOPRINT} {NORUN}
SKIP n PASS n STOP n STEP n
CMD...CP command
```

### External/Program

```
ident  typename FROM hexloc[-hexloc] {[STO raddr [ASN xxxx]]}
                                           {[ASCE raddr]}
HEX n[-m]
{EXT }
{PROG}
{TERM } {PRINT } {RUN } {SIM }
{NOTERM} {NOPRINT} {NORUN}
SKIP n PASS n STOP n STEP n
CMD...CP command
```

### Monitor Call/DIAGNOSE/SVC

```
ident  typename FROM hexloc[-hexloc] {[STO raddr [ASN xxxx]]}
                                           {[ASCE raddr]}
HEX n[-m]
{MC }
{DIAG}
{SVC }
{TERM } {PRINT } {RUN } {SIM }
{NOTERM} {NOPRINT} {NORUN} {NOSIM}
SKIP n PASS n STOP n STEP n
CMD...CP command
```

**Note:** For further response information, see [“TRACE: Command Responses”](#) on page 1954.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

## QUERY TRACEFRAMES

---

► Query — TRACEFRAMES ◄

### Authorization

Privilege Class: A, B, C, E

### Purpose

Use QUERY TRACEFRAMES to display the number of frames that are set aside for an internal trace table for the master processor and the alternate processors in the configuration.

### Usage Notes

1. CP maintains separate internal trace tables for each processor in the configuration. The QUERY TRACEFRAMES command displays the number of trace frames in the internal trace table of the master processor and the size of the internal trace tables of the alternate processors. The size of the internal trace table of an alternate processor is expressed as a percentage of the master processor's trace table.
2. Use the SET TRACEFRAMES command to change the size of the internal trace tables. For more information, see [“SET TRACEFRAMES” on page 1742](#).
3. See [“QUERY FRAMES” on page 925](#) for information on the current usage of real storage, including trace frames.

### Responses

Response 1:

If the response to a QUERY TRACEFRAMES command were:

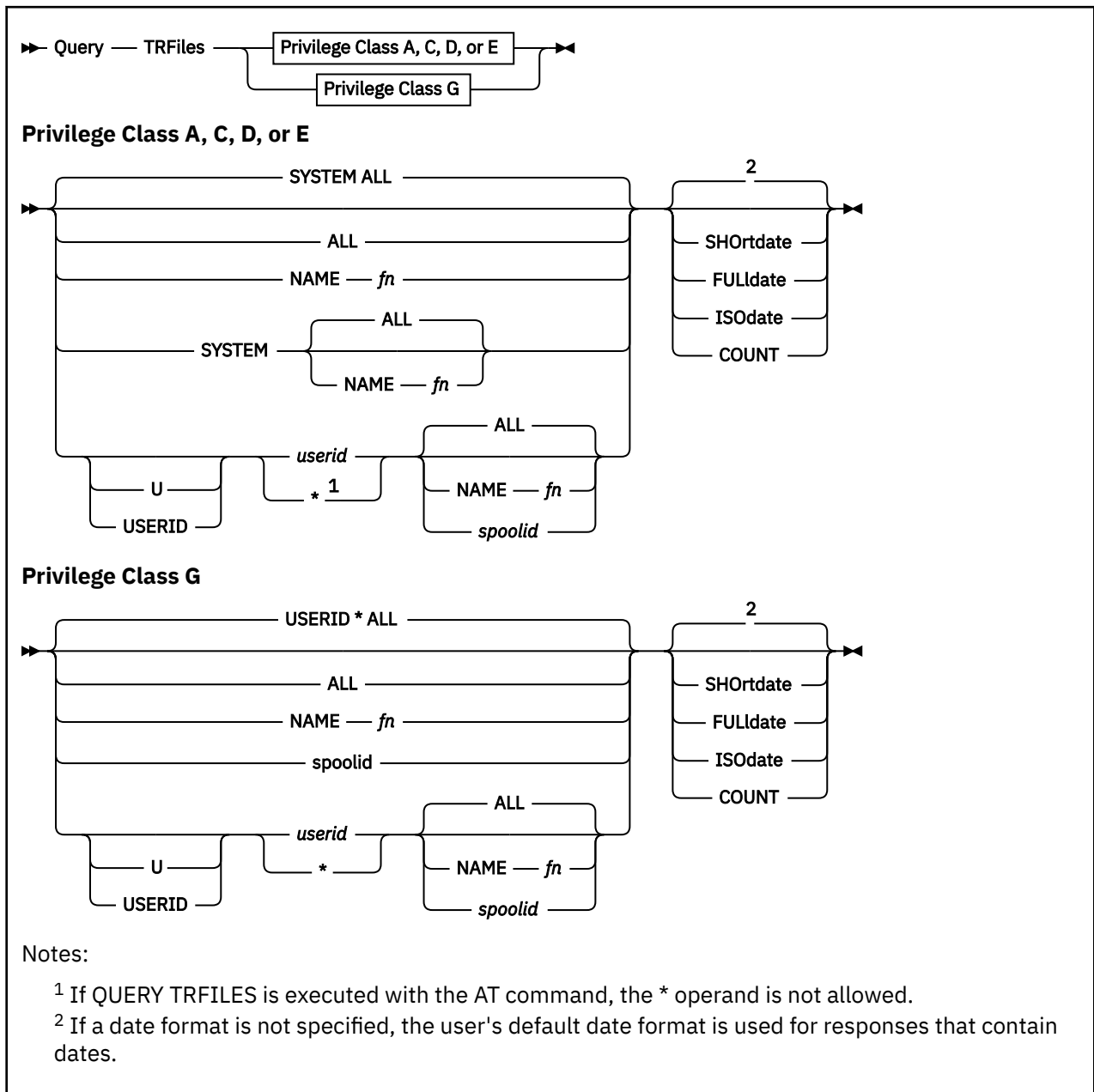
```
Master processor trace frames = 100  
Alternate processor trace frames = 90 percent of master
```

and you had a 4-processor CPU, the master processor would have a 100-frame internal trace table, and each of the three alternate processors would have a 90-frame internal trace table.

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*

# QUERY TRFILES



## Authorization

Privilege Class: A, C, D, E, G

## Purpose

Use QUERY TRFILES to display detailed information about the following:

- One or more system trace files
- The count of system trace files belonging to one or all users

**Note:** System trace files are system data files created by the TRSAVE and TRSOURCE commands.

## Operands

### ALL

queries all system trace files.

### SYSTEM

specifies that the files should be queried regardless of who owns or originated them.

### U

### USERID

indicates that the next operand is a user ID.

### \*

### *userid*

specifies the owner or originator of the files to be queried. The file owner is specified by TO *userid* on the TRSAVE command. When TO is not specified, the owner is the command issuer. For trace IDs specified by the TRSOURCE ID xxx TYPE GT FOR *userid* command, the file originator is specified by FOR *userid*. Otherwise, it is SYSTEM.

If you are a class G user, you may query only the files that you own or originate. If you, as a class G user, use the *userid* operand to specify a user ID other than your own, you receive information on all spool files that you originated and are owned by the specified user ID. If you use asterisk (\*), or you default to it, you receive information on the files you currently own.

### NAME *fn*

specifies the file name of the files to be queried. This is the file name specified on the command TRSAVE...ON DASD NAME *fn*... when the trace was started. More than one file can exist with the same file name for a given owner.

### *spoolid*

identifies the spool file to be queried.

### SHOrtdate

specifies that dates in the response to this command will be displayed in *mm/dd* format, where *mm* is the month and *dd* is the day of the month.

### FULldate

specifies that dates in the response to this command will be displayed in *mm/dd/yyyy* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

### ISOdate

specifies that dates in the response to this command will be displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

### COUNT

displays only the number of system trace files that match the criteria specified on the command.

## Usage Notes

1. The default date format for certain CP and CMS commands can be set on a system-wide basis and also for the individual user. The system-wide default date format is set with the SYSTEM\_DATEFORMAT system configuration statement. The user's default date format is set with the DATEFORMAT directory statement. The system-wide default and the user's default can also be set with the CP SET DATEFORMAT command. The user's default date format defaults to the system-wide default. The system-wide and user settings can be queried with the CP QUERY DATEFORMAT command. The hierarchy of possible date format settings for the QUERY TRFILES command, from highest priority to lowest, is:
  - QUERY TRFILES command operand
  - User default
  - System-wide default



## Responses

Response 1:

The following is an example of a response to QUERY TRFILES SYSTEM SHORTDATE:

```
OWNERID FILE TYPE CL RECS DATE TIME FILENAME FILETYPE ORIGINID
ownerid file type c recs mm/dd hh:mm:ss filename filetype
orig
```

Response 2:

The following is an example of a response to QUERY TRFILES SYSTEM FULLDATE:

```
OWNERID FILE TYPE CL RECS DATE TIME FILENAME FILETYPE ORIGINID
ownerid file type c recs mm/dd/yyyy hh:mm:ss filename filetype
orig
```

Response 3:

The following is an example of a response to QUERY TRFILES SYSTEM ISODATE:

```
OWNERID FILE TYPE CL RECS DATE TIME FILENAME FILETYPE ORIGINID
ownerid file type c recs yyyy-mm-dd hh:mm:ss filename filetype
orig
```

### OWNERID

identifies the file owner. The file owner is specified by TO *userid* when the TRSAVE command is issued; if TO is not specified, the owner is the TRSAVE command issuer.

### FILE

identifies the spool file.

### TYPE

indicates the type of system data file. This is TRF for system trace files.

### CL

identifies the class file, either A or W. If the file class is A, the file can be processed. If the file class is W, the file is in the set of files to which an active trace is writing.

### RECS

identifies the number of 4 KB pages in the file.

### DATE

identifies the date the file was closed. If the file is open for reading or writing, or if the file is being processed by the TRACERED utility, OPEN-TRSAVE is displayed instead of a date and time. The format of the date is determined by the SHORTDATE, FULLDATE, or ISODATE operand. If no date format operand is specified, the user's default date format setting is used.

### TIME

identifies the time the file was closed.

### FILENAME

identifies the file name assigned by the issuer of the TRSAVE command.

### FILETYPE

identifies the file type assigned to the file. System trace files are typed as follows:

#### Type

#### Meaning

#### CP

CP system trace tables

#### VM

Trace data for a user ID

#### VMG

Trace data for a virtual machine group

## QUERY TRFILES

### IO

Trace data from IO trace

### DATA

Trace data from DATA trace

### PCI

Trace data from PCI function trace

### ORIGINID

identifies the originator of the trace file. For traces defined by TRSOURCE...TYPE GT FOR USER... commands, *originid* is the user ID specified by FOR USER. For VM GROUPS, *originid* is the user ID that originated the trace file. For traces defined by the TRSOURCE...TYPE IO... command or by the TRSOURCE...TYPE DATA...command, *originid* is SYSTEM.

Response 4:

If no files as specified by the command line exist, the following response is given:

```
FILES:    NO TRF
```

Response 5:

If the COUNT option is specified on the QUERY TRFILES command, only the number of system trace files that match the criteria specified is displayed. The following is the response:

```
FILES: {nnnn} TRF
       { } NO{ }
```

### *nnnn*

identifies the number of system trace files. If no files are found that match the selection criteria, then NO replaces *nnnn*.

Response 6:

If an external security manager (ESM) is installed and security label checking is enabled, the ESM is called to verify the user's access to each file. If the user is denied access to a specific file, some of the information about that file must be hidden from the user. The only fields of the response that will contain information are OWNERID, FILE, TYPE, CLASS, DATE, TIME, and ORIGINID. All other fields contain asterisks.

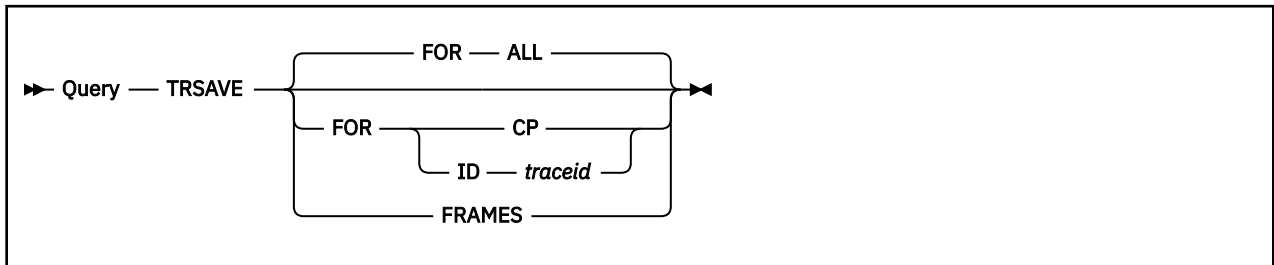
The system response when you are denied access to the file is similar to the following:

```
OWNERID  FILE TYPE CL RECS DATE  TIME      FILENAME FILETYPE ORIGINID
*****  file type c  **** date  hh:mm:ss ***** *****
orig
```

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP849E A userid must be specified with the spoolid
- HCP1558E Command processing cannot complete.

# QUERY TRSAVE



## Authorization

Privilege Class: A, C, E, G

## Purpose

Use QUERY TRSAVE to display the destination of traces defined by TRSOURCE, or to display the status of traces controlled by SET CPTRACE.

## Operands

### ALL

displays the status of all active TRSAVE functions, or indicates that there are no active functions.

### CP

displays the status of TRSAVE activity for CP system tracing. This option is allowed only for class A, C, and E users.

### FRAMES

displays the number of frames allocated from the system dynamic paging area for active DEFERIO traces and for DEFERIO CP system tracing and displays the number of pageable pages in the system dynamic paging area that are still eligible for DEFERIO traces.

### ID *traceid*

displays the status of TRSAVE activity for the specified trace ID.

## Usage Notes

1. For class G users, QUERY TRSAVE displays the destination of data derived from the trace or traces for that user's virtual machine.
2. For class A, C, or E users, QUERY TRSAVE displays the status of any TRSAVE activity.

## Responses

Response 1:

If tracing is done to DASD:

SOURCE	FILENAME	RECEIVER	KEEP	SIZE	FRAMES	FILLED	
CPTRACE	<i>filename</i>	<i>userid</i>	<i>n</i>	<i>mmm</i>	[ <i>mmmm</i>	<i>nnnn</i>	DEFERIO]
<i>traceid</i>	<i>filename</i>	<i>userid</i>	<i>n</i>	<i>mmm</i>	[ <i>mmmm</i>	<i>nnnn</i>	DEFERIO]

### SOURCE

identifies the source of the trace. CPTRACE identifies tracing activity resulting from the TRSAVE FOR CP ON command. The operand *traceid* identifies tracing activity resulting from a TRSOURCE ID *traceid* command.

## QUERY TRSAVE

### FILENAME

identifies the file names specified, or defaulted to, on the TRSAVE command.

### RECEIVER

identifies the users specified, or defaulted to, as the TO *userid* on the TRSAVE commands. Each *userid* listed in the column is the only user authorized to read this system trace file.

### KEEP

identifies the maximum number of files that CP keeps each time this trace is active.

### SIZE

identifies the maximum size, in 4 KB pages, for each of the files created when this trace is active.

### FRAMES

identifies the number of real storage frames that are set aside for an in-storage-wrap of the DEFERIO trace. The frames are allocated from the system dynamic paging area when the *traceid* is enabled or, for trace table data, when TRSAVE CP ON is issued. Blanks are displayed if the trace is not a DEFERIO trace.

### FILLED

identifies the number of those real storage frames that are filled with trace data. Blanks are displayed if the trace is not a DEFERIO trace. The *nnnnn* value will not exceed *mmmmm-1*, because one frame is set aside as the current frame being filled. When *nnnnn* reaches *mmmmm-1*, the displayed value of *nnnnn* remains at that level until the *traceid* is disabled.

### DEFERIO

indicates that no trace data is recorded on DASD until the *traceid* is disabled or, for trace table data, until TRSAVE CP OFF is issued. Blanks are displayed if the trace is not a DEFERIO trace.

When ALL is specified, the above information is displayed for all existing *traceids* and for CPTRACE when it is active.

Response 2:

The following message is displayed when a class A, C, or E user issues QUERY TRSAVE ID *traceid* for a trace ID that is not defined.

```
Traceid traceid is not defined.
```

Response 3:

The following message is displayed when a class G user issues QUERY TRSAVE ID *traceid* for a trace ID that is not defined or when the user is not authorized to view information for that particular *traceid*.

```
Traceid traceid is not tracing for your virtual machine.
```

Response 4:

The following message is displayed when the user issues QUERY TRSAVE ALL but no trace IDs have been defined.

```
No traceids defined.
```

Response 5:

The following message is displayed when a class G user issues QUERY TRSAVE ALL but no trace IDs have been defined for which the user is authorized to view information.

```
No traceids defined for your virtual machine.
```

Response 6:

If CP system trace data is being written to tape, the following message is displayed:

```
TRSAVE for CP on drive rdev1 [rdev2] MODE nnnn
```

```
{rewind} [DEFERIO FRAMES mmmmm FILLED nnnnn]  
{run }
```

**rdev1 [rdev2]**

identifies the real device number or numbers of the tape drives being used for the TRSAVE function. If only one tape drive is being used, *rdev2* is not shown.

**MODE *nnnn***

identifies the tape's format density.

**REWIND****RUN**

indicates that the tape is to be rewound (REWIND) or rewound and unloaded (RUN) when the tape fills.

**DEFERIO**

indicates that no trace data is recorded on tape until TRSAVE CP OFF is issued.

**FRAMES *mmmmm***

identifies the number of real storage frames that are set aside for an in-storage-wrap of the DEFERIO trace. The frames have been allocated from the system dynamic paging area.

**FILLED *nnnnn***

identifies the number of those real storage frames that have been filled with trace data. The *nnnnn* value will not exceed *mmmmm*-1, because one frame is set aside as the current frame being filled. When *nnnnn* reaches *mmmmm*-1, the displayed value remains at that level until TRSAVE CP OFF is issued.

Response 7:

The following message is displayed if TRSAVE FOR CP is not active:

```
TRSAVE inactive for CP
```

Response 8:

The following message indicates that TRSAVE FOR CP is neither active nor inactive, but is in a transition state between the two:

```
TRSAVE pending active for CP
```

Response 9:

The following messages are displayed when a class A, C, or E user issues QUERY TRSAVE FRAMES.

```
nnnnnn frames are allocated for active DEFERIO traces.  
mmmmmm pages are still eligible to be allocated for DEFERIO traces.
```

The frames are allocated from the system dynamic paging area. Keep in mind that number of pageable pages in the system dynamic paging area is affected by the locking and unlocking of frames for various reasons. For more information see the TRSAVE command's usage notes regarding DEFERIO.

Response 10:

The following message indicates that TRSAVE FOR CP is terminating:

```
TRSAVE terminating for CP
```

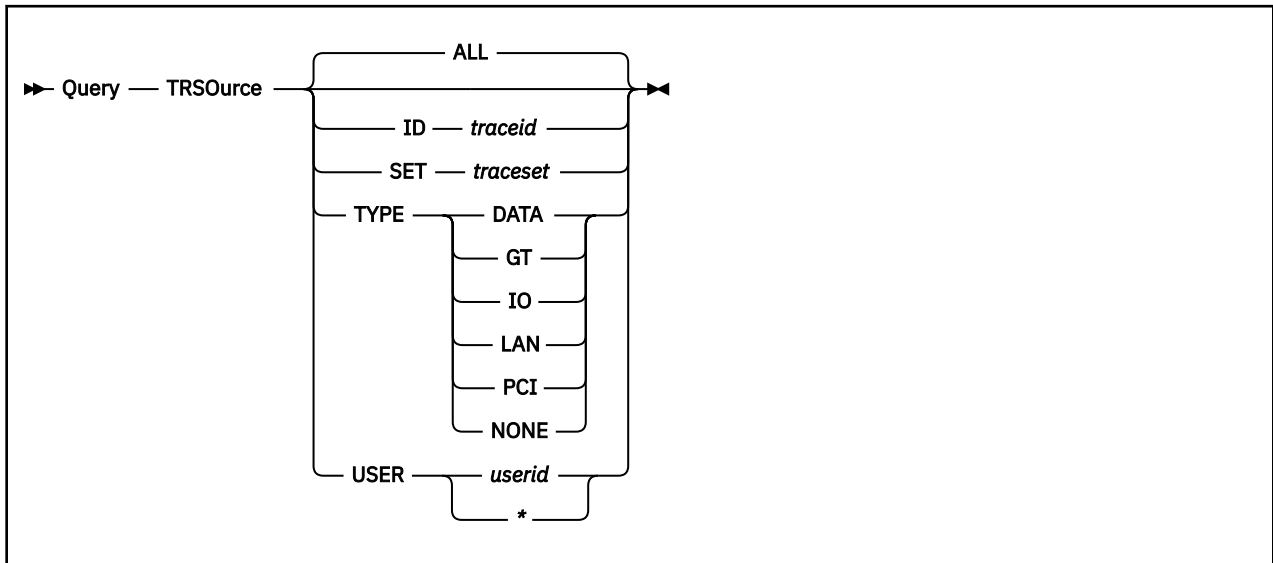
**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP410E CP ENTERED; PAGING ERROR
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*.

## QUERY TRSAVE

- HCP6525E Authorization request failed.

# QUERY TRSOURCE



## Authorization

Privilege Class: A, C, E, G

## Purpose

Use QUERY TRSOURCE to display the current status of those traces defined using TRSOURCE.

## Operands

### ALL

requests the status of all defined traces. ALL is the default.

### ID *traceid*

requests the status of the specified trace ID.

### SET *traceset*

requests the status of all defined traces of the specified set.

### TYPE DATA

### TYPE GT

### TYPE IO

### TYPE LAN

### TYPE PCI

### TYPE NONE

requests the status of all traces of the following types:

#### Type

#### Meaning

#### IO

IO

#### GT

GT

#### LAN

LAN

#### PCI

PCI

## QUERY TRSOURCE

### DATA

DATA

### NONE

Unspecified.

### USER *userid*

#### USER \*

requests the status of all traces of type GT that are defined for the specified user's virtual machine. You can use asterisk (\*) to specify your own virtual machine.

## Usage Notes

1. A trace of type NONE occurs when a trace has been defined by the TRSOURCE or TRSAVE command, but the TYPE xxx parameter of the TRSOURCE command is not specified.
2. The class G user may not specify any parameters. The response to the class G user is that of the QUERY TRSOURCE USER \* command.
3. An IO trace may be defined for a nonexistent device—that is, a device that has no RDEVICE definition. If a trace with *only* such devices is enabled, message HCP477E is issued:

```
Traceid (TRACE ID) not enabled; incomplete definition
```

Note, however, that a trace with a nonexistent device and a real device (whether the latter be online or offline) *can* be enabled, and the response to QUERY TRSOURCE indicates that a trace for the nonexistent device is activated.

## Responses for Undefined Traces

Response 1:

```
No traces are defined
```

is the response for QUERY TRSOURCE ALL if no traces are defined.

Response 2:

```
Traceid traceid is not defined
```

is the response for QUERY TRSOURCE ID trace ID if the specified trace ID is not defined.

Response 3:

```
Traceset traceset is not defined
```

is the response for QUERY TRSOURCE SET *traceset* if no traces are defined for the specified trace set.

Response 4:

```
No GT traces are defined for user userid
```

is the response for QUERY TRSOURCE USER *userid* if no GT traces are defined for the specified user's virtual machine.

Response 5:

```
No trace was defined for type type
```

is the response for QUERY TRSOURCE TYPE *type* if no traces are defined for the specified trace type.

## Responses for Defined Traces

The format of the responses depends upon the trace types. The grouping of the responses for QUERY TRSOURCE ALL is by trace set.



Response 1:

```
SET setid
```

is the response that heads each group of traces included in a set, *setid* being the set name of the set. When no set name has been defined, the set name is NULL.

Response 2:

```
ID      TYPE  SET      STATUS      DEVICES      USER      IODATA
traceid IO  setname  {ENABLED }  rdev(s)      userid    nnnn
          {DISABLED }
```

or

```
ID      TYPE  SET      STATUS      LDEVS      USER      IODATA
traceid IO  setname  {ENABLED }  ldev(s)     userid    nnnn
          {DISABLED }
```

is displayed for traces of type IO.

**ID**

identifies the name of the trace, as specified on the ID xxx parameter of the TRSOURCE...TYPE IO... command.

**TYPE**

identifies the type of trace.

**SET**

is the name used to group traces into a logically related set as specified on SET xxx parameter of the TRSOURCE ID *traceid* command or NULL.

**STATUS ENABLED**

**STATUS DISABLED**

ENABLED indicates that the defined trace is active. DISABLED indicates that the trace is inactive.

**DEVICES**

*rdev(s)* indicates the device number (*rdev1*) or ranges (*rdev1-rdev2*) of device numbers whose I/O activity should be traced. It is the value specified on the DEVICE xxx parameter of the TRSOURCE ID *traceid* command.

**LDEVS**

*ldev(s)* indicates the logical device number (*ldev1*) or ranges (*ldev1-ldev2*) of logical device numbers whose I/O activity should be traced. It is the value specified on the LDEV xxx parameter of the TRSOURCE ID *traceid* command.

**USER**

indicates which user's I/O should be traced. It is specified on the USER xxx parameter of the TRSOURCE ID *traceid* command. ALL indicates that all I/O to the specified devices are traced for all users.

**IODATA**

indicates how many bytes of the transferred data are being traced for the defined I/O trace.

Response 3:

```
ID      TYPE  SET      STATUS      ORIGINID      INCLUDED
traceid GT-x  setname  {ENABLED }  {sysname}     {userid}
          {DISABLED }  {userid}     {useridb}
```

```
ID      TYPE  SET      STATUS      ORIGINID      EXCLUDED
traceid GT-x  setname  {ENABLED }  {sysname}     {userid}
          {DISABLED }  {userid}     {useridy}
```

```
ID      TYPE  SET      STATUS      ORIGINID
traceid GT-x  setname  {ENABLED }  userid
          {DISABLED }
```

## QUERY TRSOURCE

is displayed for traces of type GT.

### ID

identifies the name of the trace, as specified on the ID *xxx* parameter of the TRSOURCE...TYPE GT... command.

### TYPE GT-x

indicates the type of the trace; *x* stands for E or B.

E indicates that the guests being traced are sending trace data to CP in EVENT mode—that is, the virtual machine sends one trace entry at a time.

B indicates that the guests being traced are sending trace data to CP in BLOCK mode—that is, the guest virtual machine is collecting the trace entries into buffers and then requesting CP to receive the entire block of trace data at a time.

### SET

is the name used to group traces into a logically related set as specified on the SET *xxx* parameter of the TRSOURCE ID *traceid* command or NULL.

### STATUS ENABLED

### STATUS DISABLED

ENABLED indicates that the defined trace is active. DISABLED indicates that the trace is inactive.

### ORIGINID

identifies a virtual machine group if TRSOURCE...FOR VMGROUP *sysname* was entered to define the GT *traceid*. When tracing is active for a virtual machine group, the user IDs of the group members for which tracing is active are displayed as shown. When TYPE GT tracing is being done for a single virtual machine (the command TRSOURCE...FOR USER *userid* was entered), *originid* is the name of the virtual machine.

### INCLUDED

displays a list of the virtual machines that are included in the virtual machine group trace. These virtual machines may or may not be logged on.

ALL indicates that every virtual machine that connects to the virtual machine group is included in the group trace.

NONE indicates that no virtual machines are included in the group trace.

### EXCLUDED

displays a list of the virtual machines that are excluded from the virtual machine group trace. These virtual machines may or may not be logged on.

**Note:** Either an included list or an excluded list is displayed for a virtual machine group.

Response 4:

```
  ID      TYPE  SET      STATUS      LOC      INSTRUCTION
  traceid DATA setname {ENABLED } { nnnnnn } string...
                   {DISABLED} {modname+disp}

DATALINKS
DATALINK 001 ...
DATALINK 002 ...
                   IF ... THEN
                   IF ... THEN
DATALINK 004 ...
                   ELSE
DATALINK 005 ...
DATALINK 006 ...
                   ENDIF
                   ENDIF
:
DATALINK XXX ...
```

- or -

```
DATALINKS
NO DATALINKS DEFINED
```

is displayed for traces of type DATA.

**ID**

identifies the trace, as specified on the ID xxx parameter of the TRSOURCE...TYPE DATA... command.

**TYPE**

indicates the type of trace.

**SET**

is the name used to group traces into a logically related set as specified on SET xxx parameter of the TRSOURCE ID *traceid* command or NULL.

**STATUS ENABLED**

**STATUS DISABLED**

ENABLED indicates that the defined trace is active. DISABLED indicates that the trace is inactive.

**LOC nnnnnn**

**LOC modname+disp**

indicates the host address of the instruction that, when executed, triggers tracing information.

The operand *modname+disp* indicates the module name and displacement of the instruction that, when executed, triggers tracing information.

**INSTRUCTION**

identifies the instruction string located at the indicated location.

**DATALINKS**

defines the information collected when the trace springs. If no data links were specified for this trace ID, NO DATALINKS DEFINED is displayed under the DATALINKS header line.

**IF**

determines whether a subsequent THEN statement or ELSE statement is executed.

**THEN**

indicates that subsequent datalinks and IF statements are executed only if the preceding IF statement is true.

**ELSE**

indicates that subsequent datalinks and IF statements are executed only if the preceding IF statement is false.

**ENDIF**

closes the last unclosed IF statement.

Response 5:

ID	TYPE	SET	STATUS
<i>traceid</i>	NONE	<i>setname</i>	DISABLED

is displayed for traces of unspecified type, or type NONE.

**ID**

identifies the name of the trace, as specified on the ID xxx parameter of the TRSOURCE command.

**TYPE**

indicates that the TYPE xxx parameter was not specified on the TRSOURCE ID *traceid* command.

**SET**

is the name used to group traces into a logically related set as specified on SET xxx parameter of the TRSOURCE ID *traceid* command.

**STATUS DISABLED**

indicates the trace is inactive.

Response 6:

ID	TYPE	SET	STATUS	OWNER	NAME	LENGTH	OPTIONS
<i>traceid</i>	LAN	<i>setname</i>	{ENABLED} {DISABLED}	<i>owner</i>	<i>lanname</i>	<i>bytes</i>	{TRUNK {DROPPED {NIC <i>owner vdev</i> {NONE

## QUERY TRSOURCE

```
VLANID
-  {vlanid(4)}
   {ALL }
```

is displayed for traces of type LAN.

### ID

identifies the trace, as specified on the ID *xxx* parameter of the TRSOURCE...TYPE LAN... command.

### TYPE

indicates the type of trace.

### SET

is the name used to group traces into a logically related set as specified on SET *xxx* parameter of the TRSOURCE ID *traceid* command or NULL.

### STATUS ENABLED

### STATUS DISABLED

ENABLED indicates that the defined trace is active. DISABLED indicates that the trace is inactive.

### OWNER *owner*

the userid that owns the specified guest LAN or virtual switch.

### NAME *lanname*

the name of the specified guest LAN or virtual switch.

### LENGTH *bytes*

the number of *bytes* of traffic to be collected.

### OPTIONS

#### TRUNK

only traffic flowing over the virtual switch OSA connection will be captured. If TRUNK is specified on a guest LAN, no traffic is collected.

#### DROPPED

only traffic that has been DROPPED due to an error or an unknown destination will be collected.

#### NIC *owner vdev*

only traffic for the specified guest machine (identified by owner and vdev) will be collected.

#### NONE

no options were specified; therefore, all traffic on the specified guest LAN or virtual switch will be captured.

### VLANID

#### *vlanid*

only traffic destined for one of the vlan ids specified will be captured. Up to four vlan ids can be specified.

#### ALL

all traffic regardless of vlan id will be collected.

Response 7:

ID	TYPE	SET	STATUS	PCIF	USER
<i>traceid</i>	PCI	<i>setname</i>	{ENABLED} {DISABLED}	<i>pcif(s)</i>	<i>userid</i>

is displayed for traces of type PCI.

### ID

identifies the name of the trace, as specified on the ID *xxx* parameter of the TRSOURCE...TYPE PCI... command.

### TYPE

indicates the type of trace.

**SET**

is the name used to group traces into a logically related set as specified on SET xxx parameter of the TRSOURCE ID *traceid* command or NULL.

**STATUS ENABLED****STATUS DISABLED**

ENABLED indicates that the defined trace is active. DISABLED indicates that the trace is inactive.

**PCIF**

*pcif(s)* indicates the PCI function (*pcif*) or ranges (*pcif1-pcif2*) of PCI functions whose activity should be traced. It is the value specified on the PCIFUNCTION xxx parameter of the TRSOURCE ID *traceid*.

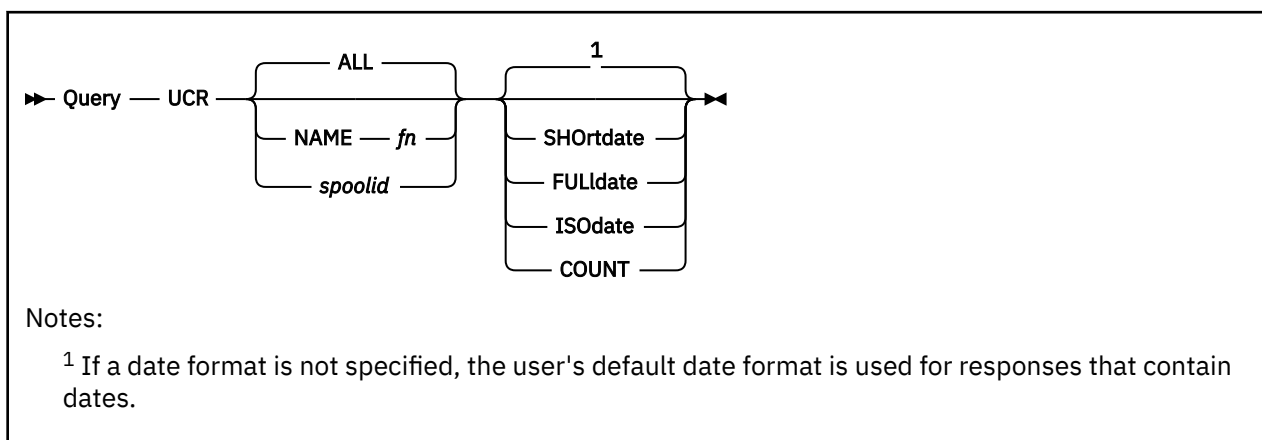
**USER**

indicates which user's PCI function should be traced. It is specified on the USER xxx parameter of the TRSOURCE ID *traceid* command. ALL indicates that activity to the specified PCI functions are traced for all users.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP6163E Internal error during command processing: *module + offset*
- HCP6525E Authorization request failed.

## QUERY UCR



### Authorization

Privilege Class: A, B, C

### Purpose

Use QUERY UCR to display detailed information about one or more user class restructure (UCR) files.

**Note:** This command is equivalent to the QUERY RDR/PRT/PUN command for spool files.

### Operands

#### ALL

lets you query all UCR files. This is the default.

#### NAME *fn*

specifies the name of the file to be queried. This is the file name specified when the file was created. More than one file can exist with the same file name.

#### *spoolid*

identifies the spool file to be queried.

#### SHOrtdate

specifies that dates in the response to this command will be displayed in *mm/dd* format, where *mm* is the month and *dd* is the day of the month.

#### FULLdate

specifies that dates in the response to this command will be displayed in *mm/dd/yyyy* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

#### ISOdate

specifies that dates in the response to this command will be displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

#### COUNT

displays only the number of UCR files that match the criteria specified on the command. No other information is provided.

### Usage Notes

1. The default date format for certain CP and CMS commands can be set on a system-wide basis and also for the individual user. The system-wide default date format is set with the SYSTEM\_DATEFORMAT system configuration statement. The user's default date format is set with the DATEFORMAT directory

statement. The system-wide default and the user's default can also be set with the SET DATEFORMAT command. The user's default date format defaults to the system-wide default. The system-wide and user settings can be queried with the QUERY DATEFORMAT command. The hierarchy of possible date format settings for the QUERY UCR command, from highest priority to lowest, is:

- QUERY UCR command operand
  - User default
  - System-wide default
2. UCR spool files and the OVERRIDE command are no longer supported. Use the MODIFY command or statement to change command privilege classes. For information about MODIFY, see [Modifying a Command and Diagnose Code in z/VM: CP Exit Customization](#). Use the CVTOVRID utility to convert a class override file (formerly processed by the CP OVERRIDE utility) into a series of MODIFY statements to imbed into a system configuration file. Use the QUERY UCR command to display information about the UCR files on your system. Use the PURGE UCR command to purge all UCR files once your conversion to MODIFY statements is complete.

## Responses

Response 1:

The following response is issued when you specify QUERY UCR SHORTDATE:

```
OWNERID  FILE TYPE CL RECS DATE   TIME
ownerid  nnnn type c  nnnn mm/dd hh:mm:ss

FILENAME FILETYPE ORIGINID
fn       ft       originid
```

Response 2:

The following response is issued when you specify QUERY UCR FULLDATE:

```
OWNERID  FILE TYPE CL RECS DATE       TIME
ownerid  nnnn type c  nnnn mm/dd/yyyy hh:mm:ss

FILENAME FILETYPE ORIGINID
fn       ft       originid
```

Response 3:

The following response is issued when you specify QUERY UCR ISODATE:

```
OWNERID  FILE TYPE CL RECS DATE       TIME
ownerid  nnnn type c  nnnn yyyy-mm-dd hh:mm:ss

FILENAME FILETYPE ORIGINID
fn       ft       originid
```

### OWNERID

indicates that the file is a UCR file owned by the system. \*UCR is not a *userid*; it indicates the queue on which the file resides.

### FILE

identifies the spool file.

### TYPE

indicates that the file is a UCR file.

### CL

identifies the class, either D or I. Class D indicates that the file was created with the DEFER option of the OVERRIDE command. Class I indicates that the file was created with the IMMEDIATE option of the OVERRIDE command.

### RECS

identifies the number of 4 KB pages in the file.

## QUERY UCR

### DATE

identifies the date that the file was closed. If the file is being created, “OPEN—UCR” is displayed instead of a date and time. The format of the date is determined by the SHORTDATE, FULLDATE, or ISODATE operand. If no date format operand is specified, the user's default date format setting is used.

### TIME

identifies the time the file was closed in hours, minutes, and seconds.

### FILENAME

identifies the file name.

### FILETYPE

identifies the file type.

### ORIGINID

identifies the file originator.

Response 4:

```
FILES:  NO UCR
```

is returned when no specified file exists.

Response 5:

If the COUNT option is specified on the QUERY UCR command, only the number of user class restructure files that match the criteria specified is displayed:

```
FILES:  {nnnn} UCR  
        { } NO }
```

### *nnnn*

identifies the number of UCR files. If no files match the selection criteria, then NO replaces *nnnn*.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP849E A userid must be specified with the spoolid
- HCP1558E Command processing cannot complete.



# QUERY UNDERSCORE

---

► Query — UNDERscore ◄

## Authorization

Privilege Class: G

## Purpose

Use QUERY UNDERSCORE to determine if an underscore character is used to separate 16-digit output values in CP command responses for your user ID.

## Usage Notes

1. To change the UNDERSCORE setting, use SET UNDERSCORE.
2. For a list of CP commands affected by the UNDERSCORE setting, see [“SET UNDERSCORE” on page 1744](#).
3. An underscore character is always allowed on input, regardless of the setting of the UNDERSCORE option. When specifying an underscore on input, there must be exactly eight characters following the underscore.

## Responses

Response 1:

```
UNDERSCORE {ON }  
           {OFF}
```

### ON

indicates that an underscore character will be used to separate 16-digit output values into two 8-digit entities.

### OFF

indicates that an underscore character will not be used to separate 16-digit output values.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*

## QUERY UNRESOLVED

► Query — UNRESolved ◄

### Authorization

Privilege Class: A, C, E

### Purpose

Use QUERY UNRESOLVED to display whether there are any references to unresolved (unknown) entry points or external symbols for customer-written CP routines that were loaded using the CPXLOAD command or configuration file statement.

### Usage Notes

1. To load customer-written CP routines into CP's virtual storage, use the CPXLOAD command (see [“CPXLOAD” on page 111](#)) or the CPXLOAD configuration file statement. For more information about the [CPXLOAD Statement](#), see *z/VM: CP Planning and Administration*.
2. The customer-written CP routines that you loaded must be on a disk that CP can access, both during and after initialization. If you have any unresolved entry points or external symbols that you thought you loaded, check to see where you loaded them. They may be on a disk that is not CP-accessed. If this turns out to be the problem, give CP access to the disk using the CPACCESS command (see [“CPACCESS” on page 89](#)).
3. To revoke one or more entry point or external symbol assignments, use the DISASSOCIATE command (see [“DISASSOCIATE” on page 324](#)).
4. To unload the customer-written CP routines, use the CPXUNLOAD command (see [“CPXUNLOAD” on page 117](#)).
5. For more information about unresolved entry points or external symbols for customer-written CP routines, see *z/VM: CP Exit Customization*.

### Responses

Response 1 (Unresolved References Exist):

To display all references that remain unresolved after all CPXLOAD requests, enter the following:

```
query unresolved
CSECT      Offset  Refers to
HCPNEW    000FBC  HCPSOON
HCPNEW    000FC0  HCPNOYET
HCPNEW    000FC4  HCPIHOPE
Ready;
```

In this example:

#### CSECT

is the control section (CSECT) of the program that was loaded using CPXLOAD. The CSECT contains address constants that refer to other entry points that CP cannot locate. Either the entry points have not yet been loaded, or they are on a disk to which CP does not have access.

#### Offset

is the location in the CSECT where the the address constant is.

**Refers to**

is the name of the entry point that the CSECT's address constant is referencing.

Response 2 (No Unresolved References):

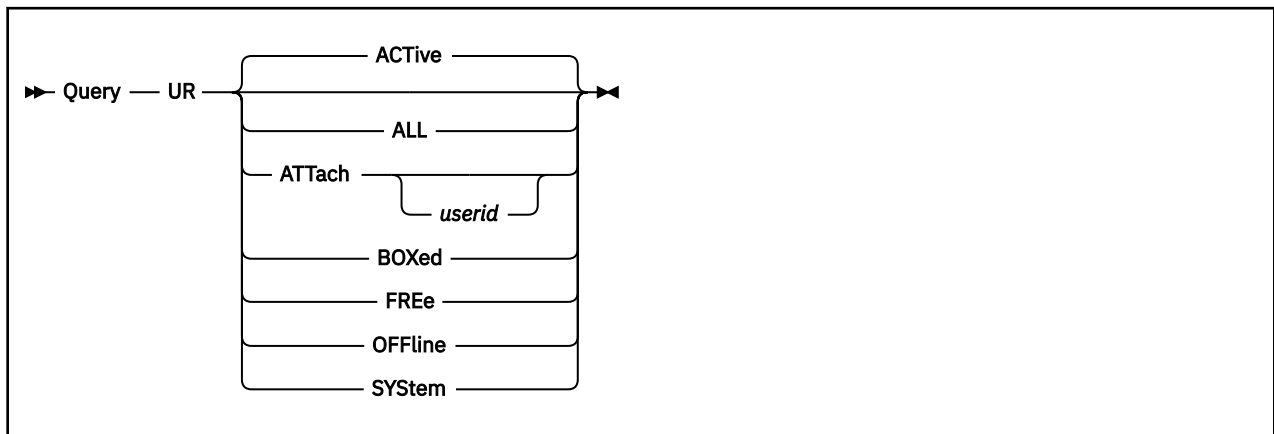
If you issue QUERY UNRESOLVED and there are no unknown entry points or external symbols, you will see:

```
query unresolved
There are no unresolved external references caused by CPXLOAD operations
Ready;
```

**Messages**

- HCP003E Invalid option - command contains extra option(s) starting with *option*

## QUERY UR



### Authorization

Privilege Class: B

### Purpose

Use QUERY UR to display the status of unit record devices.

### Operands

#### ACTive

indicates the active unit record devices. An active device is one in use by a user or the system, and specified as neither FREE nor OFFLINE. ACTIVE is the default.

#### ALL

indicates all devices of the type specified. The status is displayed in ACTIVE, FREE, and OFFLINE order, and is equivalent to the response you would receive if you entered all of the following commands:

```
query ur active
query ur free
query ur offline
```

#### ATTach *userid*

indicates the devices that are dedicated to a user on the system. If you specify *userid*, only devices attached to that user's virtual machine are indicated. If you do not specify *userid*, you receive the status of all unit record devices dedicated to any user on the system. Attached devices are always active.

#### BOXed

indicates the unit record devices that are unavailable, usually because of a hardware condition. A boxed device may also be ACTIVE, FREE, or OFFLINE.

#### FREe

indicates the free unit record devices. A free device is one not currently in use by a user or the system, and specified as neither ACTIVE nor OFFLINE.

#### OFFline

indicates the offline unit record devices. An offline device is one specified as neither ACTIVE nor FREE.

#### SYStem

indicates the unit record devices in use by the system. A system device is always active.

## Usage Notes

1. When you enter a QUERY UR command, you must make sure that the command's options are valid for each virtual machine associated with one or more *prtids*.
2. A VIRTUAL version of this command, QUERY VIRTUAL UR, also exists to facilitate a per-userid version of this data request. In some cases the VIRTUAL keyword is optional, so the output of this command may supersede QUERY VIRTUAL output. For more information, please see [“QUERY VIRTUAL UR” on page 1340](#).

## Responses

If the system is using a unit record device that is in an intervention-required state when you enter QUERY UR, you receive a response that looks like this:

```
RDR 000C has intervention required.
```

000C is the real device number of the reader that needs the operator's intervention. RDR may be replaced by PRT or PUN, depending on the type of the device.

**Note:** This response will be preceded by one of the following.

Response for Dedicated Devices:

If you enter QUERY UR and a unit record device is dedicated to a user, you see something like this:

```
q ur
RDR 000C ATTACHED TO JAMES 02F0
Ready;
```

This response means that the real card reader at the device number 000C is dedicated to JAMES's virtual machine at the virtual device number 02F0. RDR might be replaced by PRT or PUN, depending on the kind of unit record device. ATTACHED might be replaced by BOX/ATTC, which indicates that the device is attached to the virtual machine but is unavailable, usually because of a hardware problem.

Responses from Active Readers:

If a reader is assigned to the system for spooling activity and it is currently active with a spool file, you see a message like this:

```
q ur
RDR 000C STARTED SYSTEM
RDR 000C READING PERCY FILE 0021
Ready;
```

000C is the real device number of the card reader. STARTED indicates the device is available for spooling activity. SYSTEM means the device is available for input with user identification cards. STARTED might be replaced by any of the following:

- DRAINING, which would mean that the device would be unavailable for spooling activity as soon as it finished processing the current file.
- DRAINED, which would mean that the device was unavailable for spooling activity; you could enter the CP START command to restart it.
- INT REQ, which would mean that the device was not ready and required manual intervention before spool file processing could resume.

**Note:** If STARTED were replaced by any of the above options except DRAINING, the second line of the response would not appear.

SYSTEM might be replaced by a user ID, which would mean that the device was available for input without user identification cards. All input decks would be assigned to the user ID specified.

The second line of the response means that the card reader is currently active with a spool file. In this example, the reader at the real device number 000C is reading a card deck and creating a file that will be the user PERCY's spool file 0021.

Responses from Active Printers:

If there is an impact printer in use as a system spooling device, you see a response like this:

```
PRT 0E0E STARTED SYSTEM CLASS 1 SECLABEL
PRT 0E0E FORM STANDARD MANUAL SEP N03800 DEFFCB LIMIT NONE
PRT 0E0E FOLD IMAGE IMAG4248 CHARS FCB FCB8 LPP 068
PRT 0E0E DEST OFF DEST - NONE - DEST - NONE - DEST - NONE -
PRT 0E0E NOAFP
```

This response tells you the following about the impact printer at the real device number 0E0E. The first line tells you that the printer will:

- Be available for spooling activity
- Process files for any user ID
- Select only class 1 spool files for processing.

The second line tells you that it will:

- Select files requesting the STANDARD form
- Wait for the operator to schedule forms output
- Generate a separator page for output files
- Not accept spool files that contain 3800 load CCWs
- Use the default forms control buffer that was loaded into it
- Process spool files regardless of their length.

The third line tells you that it will:

- Translate (fold) lower case characters into upper case
- Use the system image library file IMAG4248
- Not use a character set
- Use the forms control buffer FCB8.
- Print separator pages with 68 lines per page.

The fourth line tells you that the printer will not use destination values as a criterion for selecting files.

The fifth line tells you that the printer will not process files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB). For more information about each line of the response, see the tables below.

You see a response like the following if there is a 3800 printer in use as a system spooling device, and that printer is currently active with a spool file:

```
PRT 0F10 STARTED SYSTEM CLASS * SECLABEL
PRT 0F10 FORM STANDARD AUTO SEP ANY3800 LIMIT NONE
PRT 0F10 FLASH 0987 IMAGE IMAG3800 CHARS GF10 FCB 6 HOLD LPP 060
PRT 0F10 DEST OFF DEST - NONE - DEST - NONE - DEST - NONE -
PRT 0F10 AFP MARK
PRT 0F10 PRINTING DAVID FILE 0002 RECS 130K COPY 005 SEQ 001
```

This response tells you the following about the printer at the real device number 0F10. The first line means that the printer will:

- Be available for spooling activity
- Process files for any user ID
- Select spool files regardless of class.

The second line tells you that the printer will:

- Select files requesting the STANDARD form
- Automatically schedule forms output
- Generate a separator page for output files
- Accept spool files that contain 3800 load CCWs
- Process spool files regardless of their length.

The third line tells you that the printer will:

- Use the forms overlay negative called 0987
- Use the system image library file IMAG3800
- Use the character set GF10
- Print six lines per inch
- Put spool files causing 3800 load checks into system HOLD status.
- Print separator pages with 60 lines per page.

The fourth line tells you that the printer will not use destination values as a criterion for selecting files.

The fifth line tells you that the printer will:

- Process files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB).
- Mark separator trailer pages with separator bars.

The last line tells you the following:

- The printer is active with the spool file 0002, owned by the user DAVID.
- The number of records remaining for output in the file is 130K, where K is the number of records in thousands rounded to the nearest 1000. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million.
- There are five copies remaining for output.
- The sequence for the file on the printer is 001.

The parts of the first line of the response for impact printers and 3800 printers might also be as follows:

Category	Possible Values	Meaning
Status	STARTED	Device is available for spooling activity.
	DRAINING	Device is processing a file, but will be drained after the file is finished.
	DRAINED	Device is not available for spooling activity. Issue the CP START command to activate the device.
	INT REQ	Device is not available, and needs manual intervention before it can continue to process spool files.
	SETUP REQ	Printer is ready to print an alignment page so that the operator can align the forms properly on an impact printer. The operator must press the STOP and START buttons to print another alignment page, or enter the START command to begin printing the entire file.
	MOUNT REQ	Printer is waiting for the operator to mount a new form.
User ID	SYSTEM	Device processes files for any user ID.
	<i>userid</i>	Device processes files only for the specified user ID.

Category	Possible Values	Meaning
Spool file class	CLASS <i>cl</i>	Device services spool files of the specified classes, in the specified order. An output device can service as many as 8 output classes.

The parts of the second line of the response can be:

Category	Possible Values	Meaning
Form	FORM <i>form</i>	Device services the specified 1- to 8-character operator form number.
Scheduling	MANUAL	Operator schedules forms output. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is printed, the printer stops.
	AUTO	Operator wants CP to schedule forms output. The form that is currently active is processed first. When all files with the current form have been printed, CP selects a new form for the device.
	SETUP	Operator wants CP to print a page for forms alignment verification. (Only valid for impact printers.) The operator receives a message to set up the form. After setting up the form, the operator must press the STOP button, then the START button. An alignment page prints with letters replaced by X's and numbers replaced by 9's on the first page of the file.  Each time the operator presses STOP and START after a full alignment page is printed, another alignment page prints. After aligning the forms, the operator enters the START command for the device. The printer then returns to its previous mode (AUTO or MANUAL), and the file prints in its entirety.  <b>Note:</b> If neither MANUAL, AUTO, nor SETUP is specified and the printer is currently in SETUP mode, it returns to the previous mode, AUTO or MANUAL. The default forms output is MANUAL.
File separator	SEP	File separator prints between output files.
	NOSEP	File separator does not print between output files.
3800 load CCWs	NO3800	Printer does not print any spool files containing 3800 load CCWs. This option ensures a fast output queue, because load CCWs issued to 3800 printers can take a lot of time.
	BEG3800	Only spool files without 3800 load CCWs, or spool files with 3800 load CCWs only at the beginning of the file, print on this device. This allows the user an initial setup with which the entire file can be printed.
	ANY3800	Output device will process spool files regardless of whether they contain 3800 load CCWs. This option may cause a slower output queue, based on the number of 3800 load CCWs issued by the spool files.



Category	Possible Values	Meaning
Default FCB	DEFFCB	Printer uses the default FCB with which it was loaded. CP does not send imbedded LOAD FCB CCWs to the printer. The LOADBUF command specifies the default FCB.
	FILEFCB	CP sends FCBs to the real printer. File FCBs are LOAD FCB CCWs imbedded in the spool file. This is the default when you start the printer after a cold start. <b>Note:</b> DEFFCB and FILEFCB only appear for non-1403 impact printers.
Length limit	LIMIT NONE	CP selects files for printing regardless of the number of records.
	LIMIT <i>nnnnnnnnn</i>	CP only selects spool files for printing that are less than or equal to the specified number of records. The spool file record number is the number of copies multiplied by the size of the file.

The parts of the third line of the response can be:

Category	Possible Values	Meaning
Folding into upper case	FOLD	CP translates (folds) lower case characters into upper case.
	NOFOLD	CP does not translate lower case characters into upper case. <b>Note:</b> FOLD and NOFOLD only appear for impact printers.
Forms overlay negative	FLASH <i>name</i>	Printer uses the specified forms overlay negative, which is from 1 to 4 characters long. <b>Note:</b> FLASH only appears for 3800 printers.
Image library	IMAGE <i>imagelib</i>	Printer uses the specified system image library file, which contains the load images (character sets, FCBs, graphic character modifications, or copy modifications) used for printing files and separator pages.
Character set	CHARS <i>chars</i>	Printer uses the specified character set, which is from 1 to 4 characters long, when printing the separator page.
FCB	FCB <i>name</i>	For impact printers: printer uses the specified FCB when printing the file or the separator page. For 3800 printers: the FCB is the number of lines per inch. <b>Note:</b> FCB does not appear for 1403-type printers.
3800 load checks	HOLD	Printer puts spool files causing 3800 load checks in system HOLD status.
	PURGE	Printer deletes spool files causing 3800 load checks. <b>Note:</b> HOLD and PURGE only appear for 3800 printers.
Lines per page	LPP	Defines the number of lines for file separator pages on the printer.

The parts of the fourth line of this response can be:

Category	Possible Values	Meaning
Destination	DEST	Printer services the specified destination values. The printer can service up to 4 destination values. The first value for the device is always OFF if it was not specifically set by either a START command, or at initialization with the RDEVICE statement in the system configuration file. If it was set, the specified destination is shown. If the remaining destination values were not specified, the values are null and are shown as – NONE –. If they were specified, the specified value is shown.
AFP characteristics	AFP	Printer processes files with advanced function printer (AFP) characteristics, that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB).
	NOAFP	Printer does not process files with AFP characteristics.
Mark separator trailer pages	MARK	Printer marks 3800 trailer pages with separator bars.
	NOMARK	Printer does not mark 3800 trailer pages with separator bars.

**Note:** MARK and NOMARK only appear for 3800 printers.

Responses from Active Card Punches:

If a punch is attached to a user and is currently active with a spool file, you see a message like:

```
PUN 1F30 STARTED   SYSTEM CLASS 1
PUN 1F30 FORM STANDARD AUTO   NOSEP LIMIT NONE
PUN 1F30 DEST OFF   DEST - NONE - DEST - NONE - DEST - NONE -
PUN 1F30 PUNCHING JAMES   FILE 0042 RECS 012K COPY 002 SEQ 001
```

This response tells you the following about the card punch at the real device number 1F30. The first line means that the punch will:

- Be available for spooling activity
- Process files for any user ID
- Select only class 1 spool files

The second line tells you that the punch will:

- Select files requesting the STANDARD form
- Automatically schedule forms output
- Not generate a separator page for output files

The third line tells you that the punch will not use destination values as a criterion for selecting files.

The parts of the first line of the response can also be:

Category	Possible Values	Meaning
Status	STARTED	Device is available for spooling activity.
	DRAINING	Device is processing a file, but will be drained when the file is finished.
	DRAINED	Device is not available for spooling activity. Issue the CP START command to activate the device.
	INT REQ	Device is not available; it needs manual intervention before it can continue to process spool files.
	SETUP REQ	Punch is ready to generate an alignment page to let the operator align the forms properly on an impact printer. The operator must press the STOP and START buttons to print another alignment page, or enter the START command to begin printing the entire file.
	MOUNT REQ	Punch is waiting for the operator to mount a new form.
User ID	SYSTEM	Device processes files for any user ID.
	<i>userid</i>	Device processes files only for the specified user ID.
Spool file class	CLASS <i>cl</i>	Device services spool files of the specified classes, in the specified order. An output device can service as many as 8 output classes.

The options for the second line of the response can be:

Category	Possible Values	Meaning
Form	FORM <i>form</i>	Device services the specified 1- to 8-character operator form number.
Scheduling	MANUAL	Operator schedules forms output. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is punched, the punch stops.
	AUTO	Operator wants CP to schedule forms output. The form that is currently active is processed first. After punching all files with the current form, CP selects a new form for the device.
File separator	SEP	File separator prints between output files.
	NOSEP	File separator does not print between output files.
Length limit	LIMIT NONE	CP selects files for punching regardless of the number of records.
	LIMIT <i>nnnnnnnnnn</i>	CP only selects spool files for punching that are less than or equal to the specified number of records. The spool file record number is the number of copies multiplied by the size of the file.

The options for the third line of the response can be:

Category	Possible Values	Meaning
Destination value	DEST	Punch services the specified destination values. The punch can service up to 4 destination values. The first value for the device is always OFF if it was not specifically set by either a START command, or at initialization with the RDEVICE statement in the system configuration file. If it was set, the specified destination is shown. If the remaining destination values were not specified, the values are null and are shown as – NONE –. If they were specified, the specified value is shown.

The last line of this response only appears if there is a file active on the card punch. The line in the example above tells you that the punch at the real device number 1F30 is punching the spool file 0042, owned by the user JAMES. The number of records remaining for output in the file is 12K, where K is the number of records in thousands rounded to the nearest 1000. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines in millions rounded to the nearest million. There are two copies remaining for output. The sequence for the file on the punch is 001.

Response from Boxed Unit Record Devices:

If a unit record that is not dedicated to a user is unavailable, you see a message like the following:

```
q ur
RDR 000F BOXED
Ready;
```

000F is the real device number of the card reader that is unavailable. RDR might be replaced by PRT or PUN.

Response from CP if there is no match:

If CP cannot find any unit record devices of the type you specified, you see a message like this:

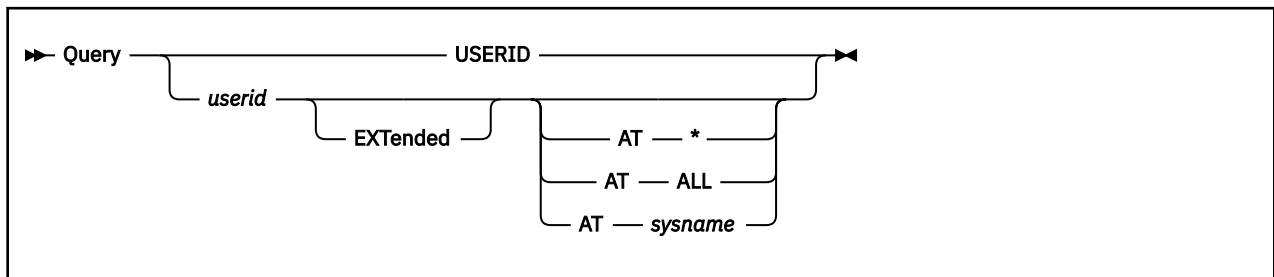
```
q ur boxed
A BOXED unit record device was not found.
Ready;
```

BOXED may be replaced with *active*, *attached*, *free*, *offline*, or *system*, depending on what you specified on the command.

## Messages

- HCP003E Invalid option - *option*
- HCP007E Invalid userid - *userid*
- HCP263E Too many operands were supplied for this command.

## QUERY USERID



### Authorization

Privilege Class: G

### Purpose

Use QUERY USERID to identify the user ID and the system name or device address of a logged-on user.

### Operands

#### USERID

displays the user ID and local system name for the user issuing the command.

#### *userid*

displays the user ID and device address of the specified user.

#### EXTENDED

specifies that logical unit network qualifiers or logical device host names and associated IP addresses are displayed, if they exist. In an SSI cluster, the member where the specified user is logged on is displayed. See [Response 3](#) for details on how the EXTENDED operand changes the responses.

#### AT \*

displays the user ID and device address of the specified user only if that user is logged on to the local system.

#### AT ALL

displays the user ID and device address of the specified user on all members of the SSI cluster. The command issues a response for each member where the user is logged on. The responses are displayed in the order they are received, and each response is prefixed with the source member name.

#### AT *sysname*

displays the user ID and device address of the specified user on system *sysname*, which must be the local system or another member of the same SSI cluster.

### Usage Notes

1. In an SSI cluster, a remote single-configuration virtual machine or any local user might be returned in response to a QUERY *userid* command. If the user is remote, "SSI" is displayed instead of a device address. To get the device address and member name for a remote single-configuration virtual machine, issue QUERY *userid* EXTENDED. To get information about all the logon instances of a multiconfiguration virtual machine, issue QUERY *userid* AT ALL.

### Responses

Response 1:

When you issue QUERY USERID, you will receive the following response:

## QUERY USERID

```
userid [AT sysname]
```

### **sysname**

is the 1- to 8-character name of the system where the command is issued. If no name has been assigned to the system being used, this field is omitted.

Response 2:

When you issue `QUERY userid`, you will receive the following response:

```
userid - {rdev }
         {DSC }
         {SSI }
         {ldev }
         {luname}
         {SYSC}
         {SYSG }
```

### **userid**

identifies the user specified.

### **rdev**

identifies the real device number of the display where the user is logged on.

### **DSC**

identifies a user disconnected from the system.

### **SSI**

identifies a user logged on to another member of the SSI cluster.

### **ldev**

identifies the logical device number through which CP accesses the display where the user is logged on.

### **luname**

identifies the SNA logical unit name of the terminal through which the user is logged on.

### **SYSC**

identifies the system console.

### **SYSG**

identifies the integrated 3270 console.

Response 3:

When you issue `QUERY userid EXTENDED`, you will receive the following response:

```
userid - {rdev [AT member] }
         {DSC [AT member] }
         {ldev HOST hostname [FROM ipaddr [AT member]] }
         {[netname.]luname [AT member] }
         {SYSC [AT member] }
         {SYSG [AT member] }
```

### **AT member**

identifies the member of the SSI cluster that the user is logged on to.

### **HOST hostname**

identifies the virtual machine that created and manages the logical device where the user is logged in.

### **FROM ipaddr**

provides the IP address of the TCP/IP Telnet client associated with the logical device.

An IPv4 address is displayed in dotted-decimal format, for example:

```
9.130.44.144
```

An IPv6 address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros). For example, this IPv6 address:

```
F802:0000:0000:0000:0000:0000:0A00:0001
```

is displayed in compressed format as:

```
F802::A00:1
```

### ***netname.luname***

identifies the fully qualified SNA logical unit name of the terminal through which the user is logged on.

Response 4:

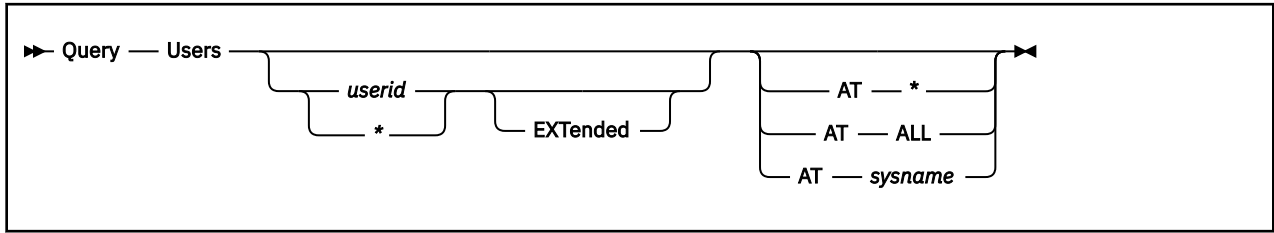
When you issue QUERY *userid* AT ALL in an SSI cluster, you will receive the following response:

```
member : userid - {rdev }
                  {DSC }
                  {ldev }
                  {luname}
                  {SYSC }
                  {SYSG }
:
```

### **Messages**

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP045E *userid* not logged on
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.

## QUERY USERS



### Authorization

Privilege Class: Any

### Purpose

Use QUERY USERS to display:

- The number of logged-on users
- The number of users dialed to other virtual machines
- The device number of a specified user

### Operands

#### *userid*

displays the user ID and device address of the specified user.

**\***

displays the user ID and device address of the user who issues the command.

#### **EXTended**

specifies that logical unit network qualifiers or logical device host names and IP addresses are displayed, if they exist. In an SSI cluster, the member where the specified user is logged on is displayed. See [Response 3](#) for details on how the EXTENDED operand changes the responses.

#### **AT \***

displays the number of users logged on to the system you are logged on to. When specified with *userid*, it displays the user ID and device address of that user on the system you are logged on to.

#### **AT ALL**

displays the number of users logged on to all members of the SSI cluster. When specified with *userid*, it displays the user ID and device address of that user on all members of the SSI cluster.

If *userid* is not specified, the command issues a response for each active member. If *userid* is specified, the command issues a response for each member where that user is logged on. The responses are displayed in the order they are received, and each response is prefixed with the source member name.

#### **AT sysname**

displays the number of users logged on to system *sysname*. When specified with *userid*, it displays the user ID and device address of that user on system *sysname*. The specified system must be the local system or another member of the same SSI cluster.

### Usage Notes

1. A dialed display device is not available to CP because it is logically attached to a logged-on user and is a part of that user's virtual machine configuration. For more information, see [“DIAL” on page 308](#).
2. In an SSI cluster, if QUERY USERS is issued with no additional operands, the response displays values that are sums for the entire cluster. If counts from only the local system are desired, issue QUERY



USERS AT \*. If counts for each member are desired, issue QUERY USERS AT ALL. User counts for other cluster members are cached and updated each time a virtual machine logs on or off.

3. In an SSI cluster, a remote single-configuration virtual machine or any local user might be returned in response to a QUERY USERS *userid* command. If the user is remote, "SSI" is displayed instead of a device address. To get the device address and member name for a remote single-configuration virtual machine, issue QUERY USERS *userid* EXTENDED. To get information about all the logon instances of a multiconfiguration virtual machine, issue QUERY USERS *userid* AT ALL.
4. QUERY NAMES and QUERY USERS can return different responses for the number of users logged on. The number of users in the process of logging on and logging off accounts for this difference. Also, in an SSI cluster, QUERY NAMES does not include multiconfiguration virtual machine logon instances on all members of the cluster (that is, the user ID is displayed once), but they are included in the totals displayed by QUERY USERS.

## Responses

Response 1:

When you issue QUERY USERS, you will receive the following response:

```
nnnnnn USERS, mmmmmm DIALED, xxxxxx NET
```

### **nnnnnn**

is the total number of logged-on users.

### **mmmmmm**

is the total number of users logically attached to virtual machines using the DIAL command.

### **xxxxxx**

is the total number of SNA/CCS terminals currently in use. This number includes SNA/CCS terminals that are at the logon prompt or are in use by either a LOGON or DIAL connection.

Response 2:

When you issue QUERY USERS *userid*, you will receive the following response:

```
userid - {rdev }
         {DSC }
         {SSI }
         {ldev }
         {luname}
         {SYSC }
         {SYSG }
```

### **userid**

identifies the user specified.

### **rdev**

identifies the real device number of the display where the user is logged on.

### **DSC**

identifies a user disconnected from the system.

### **SSI**

identifies a user logged on to another member of the SSI cluster.

### **ldev**

identifies the logical device number through which CP accesses the display where the user is logged on.

### **luname**

identifies the SNA logical unit name of the terminal through which the user is logged on.

### **SYSC**

identifies the system console.

## QUERY USERS

### SYSG

identifies the integrated 3270 console.

Response 3:

When you issue `QUERY USERS userid EXTENDED`, you will receive the following response:

```
userid - {rdev [AT member]
         {DSC [AT member]
         {ldev HOST hostname [FROM ipaddr [AT member]]
         {[netname.]luname [AT member]
         {SYSC [AT member]
         {SYSG [AT member]
```

### AT member

identifies the member of the SSI cluster that the user is logged on to.

### HOST hostname

identifies the virtual machine that created and manages the logical device where the user is logged in.

### FROM ipaddr

provides the IP address of the TCP/IP Telnet client associated with the logical device.

An IPv4 address is displayed in dotted-decimal format, for example:

```
9.130.44.144
```

An IPv6 address is displayed in compressed format (":" designates multiple 16-bit groups of zeros). For example, this IPv6 address:

```
F802:0000:0000:0000:0000:0A00:0001
```

is displayed in compressed format as:

```
F802::A00:1
```

### netname.luname

identifies the fully qualified SNA logical unit name of the terminal through which the user is logged on.

Response 4:

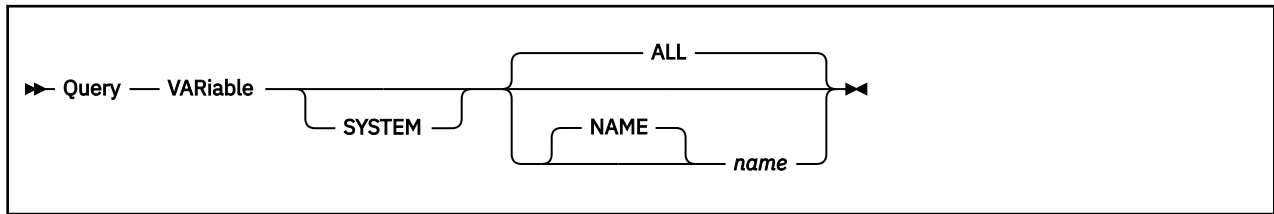
When you issue `QUERY USERS userid AT ALL` in an SSI cluster, you will receive the following response:

```
member : userid - {rdev }
          {DSC }
          {ldev }
          {luname}
          {SYSC }
          {SYSG }
:
```

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP045E *userid* not logged on
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.

## QUERY VARIABLE



### Authorization

Privilege Class: G

### Purpose

Use QUERY VARIABLE to obtain a list of system environment variables or the value of a single variable.

### Operands

#### SYSTEM

indicates that system environment variables are to be displayed.

#### ALL

displays all of the environment variables in alphabetical order.

#### NAME

displays the value of the variable with the specified name.

#### *name*

specifies the name of the environment variable, which must satisfy the following rules:

- It is 1 - 63 characters in length with no blanks.
- Valid characters are 0-9, A-Z, \_ (underscore), and . (period).
- It cannot begin or end with a period
- It cannot begin with a numeric digit.

### Usage Notes

1. The **QUERY VARIABLE** responses are expected to be read by programs and are designed to be easily parsed by programs and read by users, as follows:
  - a. No extra blanks or headers are included.
  - b. When a single variable is queried, one output line is provided that displays only the variable's value. The variable name is not included in the response.
  - c. The query of an undefined variable does not generate an error; it displays a null line, or null string. A *null string* is a string that contains no characters. Null strings have a length of zero.
  - d. When **QUERY VARIABLE SYSTEM ALL** is issued, one output line is provided for each defined variable. Each line begins with SYSTEM, and is followed by the variable name and value, with a single blank (x'40') between SYSTEM, name, and value.
  - e. Output lines are wrapped on the display screen according to the TERMINAL LINESIZE, but are not split in the Diagnose x'08' response buffer. When returning the response in a buffer, Diagnose x'08' processing adds an x'15' line-end character at the end of each response line.
2. To avoid potential ambiguity and ensure upward compatibility, especially when using this command in a program, code the ALL or NAME keyword explicitly.

## QUERY VARIABLE

3. Be careful when using **QUERY VARIABLE NAME** *name* with the cp built-in program of CMS Pipelines, as in this REXX statement:

```
'PIPE CP QUERY VARIABLE NAME RUNMODE | VAR rmode'
```

The cp program discards null lines in the CP command response. Therefore, if RUNMODE is an undefined CP environment variable, the stage emits zero records, rather than emitting a single record of length zero. Consequently, the REXX variable *rmode* will be undefined following execution of this statement, rather than having the null string as its value. The REXX function Symbol(*rmode*) or the Signal On NoValue statement can be used to detect this condition, or a technique like the following could be used to set *rmode* to the default value 'NORMAL' when RUNMODE is undefined:

```
'PIPE CP QUERY VARIABLE NAME RUNMODE|APPEND LITERAL NORMAL|VAR rmode'
```

4. Any variable with the prefix "CP." is interpreted by CP as an attribute of the system, configuration, or environment. For example, CP.IPLPARMS.IPLVAR is set according to an option in the IPL parameters, and variables of the form CP.FUNCTION.xxx indicate the level of functionality within the CP nucleus. A full list of valid variable names and values as well as descriptions of those variables can be found at the following location: <https://www.vm.ibm.com/newfunction/varlist.html>
5. You can use a CMS pipeline to filter the **QUERY VARIABLE** output for specific groups of variables. For example, to select CP new function variables, use the following REXX statement:

```
'pipe CP QUERY VARIABLE | strfind /SYSTEM CP.FUNCTION./ | console'
```

## Responses

Response 1:

When the following command is issued to set an environment variable:

```
SET VARIABLE SYSTEM RUNMODE NORMAL
```

the response to **QUERY VARIABLE ALL** is:

```
SYSTEM RUNMODE NORMAL
```

Response 2:

When the following command is issued to set an environment variable:

```
SET VARIABLE SYSTEM RUNMODE NORMAL
```

the response to **QUERY VARIABLE RUNMODE** is:

```
NORMAL
```

Response 3:

When the following commands are issued to set environment variables:

```
SET VARIABLE SYSTEM RUNMODE NORMAL
SET VARIABLE SYSTEM TESTLEVEL XR_TRIAL1
SET VARIABLE SYSTEM K.GROUP 1TSR
```

the response to **QUERY VARIABLE ALL** is:

```
SYSTEM K.GROUP 1TSR
SYSTEM RUNMODE NORMAL
SYSTEM TESTLEVEL XR_TRIAL1
```

Response 4:

When the following commands are issued to set environment variables:

```
SET VARIABLE SYSTEM RUNMODE NORMAL
SET VARIABLE SYSTEM TESTLEVEL XR_TRIAL1
SET VARIABLE SYSTEM K.GROUP 1TSR
```

the response to **QUERY VARIABLE NAME TESTLEVEL** is:

```
XR_TRIAL1
```

Response 5:

When the following command is issued to delete an environment variable:

```
SET VARIABLE SYSTEM TEMP_LIMIT
```

the response to **QUERY VARIABLE NAME TEMP\_LIMIT** is a null line.

Response 6:

This example shows how embedded blanks can be used in the environment variable value. A sample REXX EXEC, which sets a long environment variable value, follows.

When the following is issued to set environment variables:

```
(a sample REXX exec)
/* This comment format indicates the REXX interpreter is to be used */
address command
"CP SET VARIABLE SYSTEM RUNMODE NORMAL"
"CP SET VAR SYSTEM ERROR_LOG7      First I Logged on,      then",
"I brought up my system and issued the command START TEST3 then ",
"I got the MESSAGE 'XYZ ABEND43    TRY AGAIN LATER  '"
```

The response to **QUERY VARIABLE ALL** is shown below as it is displayed on a terminal screen when **TERMINAL LINESIZE 60** is set:

```
SYSTEM ERROR_LOG7 First I Logged on,      then I brought up m
y system and issued the command START TEST3 then I got th
e MESSAGE 'XYZ ABEND43    TRY AGAIN LATER  '
SYSTEM RUNMODE NORMAL
```

## Messages

- HCP0002E Invalid operand - *operand*
- HCP0003E Invalid option - *{option | command contains extra option(s) starting with option}*
- HCP1001E An operand is missing for NAME.
- HCP2148I No environment variables exist.
- HCP2148I No system environment variables exist.
- HCP6706E Invalid variable name - *name*

## QUERY VCONFIG

---

► Query — VCONFIG ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY VCONFIG to display the virtual configuration mode for a virtual machine.

### Usage Notes

1. The SET VCONFIG command can be issued to change the virtual configuration mode for the virtual machine.

### Responses

Response 1:

```
MODE = m
```

where:

***m***

identifies the configuration mode and is set to one of the following:

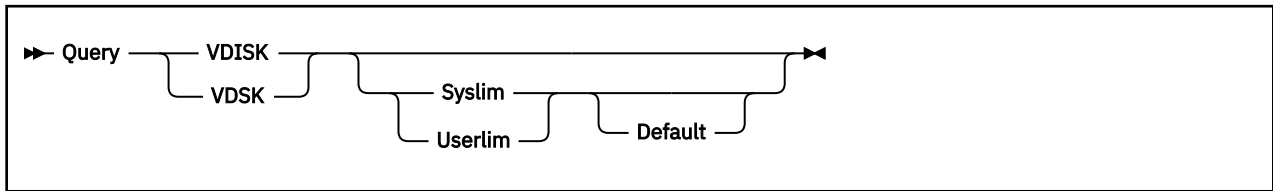
*ESA390*  
*GENERAL*  
*LINUX*  
*VM*

VCONFIG modes ESA390 and GENERAL are functionally equivalent, which means that the same virtual CPU types can be defined in an ESA390 mode virtual machine and in a GENERAL mode virtual machine. For the SET VCONFIG command, ESA390 and GENERAL are synonyms and cause a configuration of the same capabilities to be defined when specified. The QUERY VCONFIG command will display ESA390 when MACHINE is set to XA, ESA, or XC, and will display GENERAL when MACHINE is set to Z.

### Messages

- HCP263E Too many operands were supplied for this command.

## QUERY VDISK



### Authorization

Privilege Class: B

### Purpose

Use QUERY VDISK to display a list of all existing virtual disks in storage, or to display information about the system or user limits on the storage available for virtual disks in storage.

### Operands

#### Syslim

displays the maximum amount of system storage that may be allocated for virtual disks in storage on the system.

#### Userlim

displays the maximum amount of system storage that may be allocated for virtual disks in storage defined by a single user using the DEFINE command.

#### Default

displays the default limit. If an installation default is defined in the system configuration file, that value is displayed. If an installation default is not defined in the system configuration file, or if the file is not found, the built-in default is displayed.

### Responses

Response 1:

If QUERY VDISK or QUERY VDSK is entered with no additional operands:

```
VDISK userid vdev nnnnnnnn BLK [nnnn LINKS]
```

is displayed for each virtual disk in storage that currently exists.

#### *userid*

is the owner of the virtual disk in storage.

#### *vdev*

is the virtual device number of the virtual disk in storage in the owner's virtual machine.

#### *nnnnnnnn* BLK

is the size of the virtual disk in storage in 512-byte blocks.

#### *nnnn* LINKS

is the number of links to this virtual disk in storage. This value is displayed only for a virtual disk in storage defined in the directory.

Response 2:

```
userid vdev {R/O}, ...  
          {R/W}
```

## QUERY VDISK

is displayed following Response 1 for each current link to the virtual disk in storage. Up to three responses are displayed on each line.

### ***userid***

is the user ID of the linking user.

### ***vdev***

is the virtual device number of the virtual disk in storage in the linking user's virtual machine.

### **R/O**

### **R/W**

identifies the type of access the linking user has to the virtual disk in storage.

Response 3:

```
VDISKS NOT FOUND
```

is displayed for a QUERY VDISK or QUERY VDSK command with no additional operands if no virtual disks in storage are in use.

Response 4:

```
VDISK SYSTEM LIMIT IS {nnnnnnnnnn BLK,} nnnnnnnnnn  
                      {INFINITE,  }  
BLK IN USE
```

is displayed if the command is entered with the SYSLIM operand.

Response 5:

```
VDISK USER LIMIT IS {nnnnnnnnnn BLK}  
                   {INFINITE  }
```

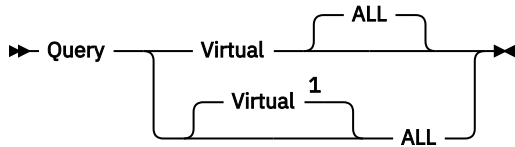
is displayed if the command is entered with the USERLIM operand.

## **Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - [command contains extra option(s) starting with] *option*
- HCP026E Operand missing or invalid



## QUERY VIRTUAL ALL



Notes:

<sup>1</sup> This operand is required for privilege class G users who also have privilege class B.

### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL ALL to display the size of your storage and the status of your devices and processors.

### Operands

#### ALL

displays the status of all devices in your virtual machine system, the size of your virtual storage, and the status of your virtual processors. This is the default.

### Usage Notes

1. The response order for this command is as follows:

- Storage
- Expanded Storage
- CPUs
- Consoles
- Readers
- Punches
- Printers
- DASDs
- FCPs
- ASCII console
- Tapes
- Displays
- Lines
- CTCAs
- OSAs
- CTLR
- Switches
- Message devices
- Unsupported devices.

## QUERY VIRTUAL ALL

- Virtual cryptographic facilities

### Responses

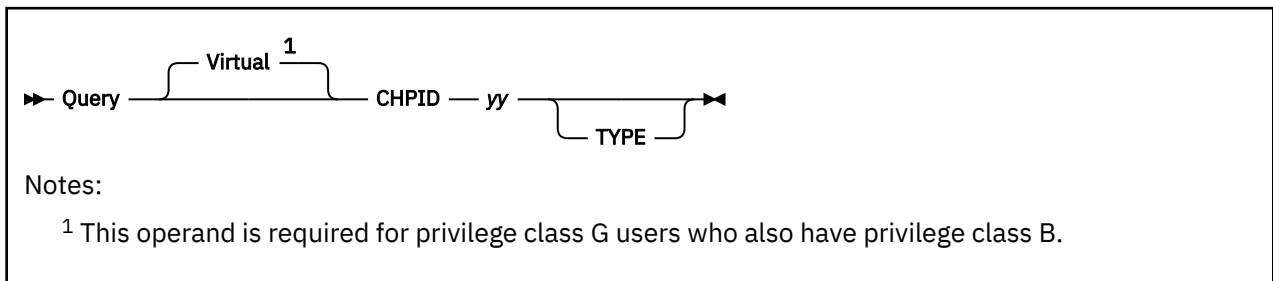
You receive responses for all devices that are defined in your virtual machine system. These responses are explained in the QUERY VIRTUAL command corresponding to each device.

**Note:** Expanded storage is not supported. The response XSTORE=- - - is displayed only for compatibility.

### Messages

- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command.
- HCP1003E {*CP module name/rdev/ldev*} {*is|was*} not found.

## QUERY VIRTUAL CHPID



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL CHPID to display the devices associated with a virtual CHPID number or the virtual CHPID type.

### Operands

#### CHPID

*yy* is the logical channel path of the virtual CHPID whose device allocation information is requested. The variable *yy* must be a 1- to 2-digit hexadecimal number between X'00' and X'FF'.

#### TYPE

Display the CHPID type.

### Responses

Response for Path Status:

```
PATH yy {ONLINE } TO DEVICES xxxx(1)
... xxxx(n)

PATH yy {OFFLINE} TO DEVICES zzzz(1)
... zzzz(n)
```

indicates that path *yy* to device *xxxx*(1) through device *xxxx*(*n*) is logically available (online), and path *yy* to device *zzzz*(1) through device *zzzz*(*n*) is logically unavailable (offline).

#### *yy*

is the logical channel path.

#### *xxxx*

#### *zzzz*

are the real device numbers.

Response for No Devices on Path:

```
NO DEVICES ON PATH yy
```

indicates there are no devices on the specified channel path.

#### *yy*

is the logical channel path.

Response for Nonexistent CHPID:

```
INVALID CHPID yy
```

indicates that a nonexistent channel path ID was specified.

**yy**

is the nonexistent logical channel path.

Responses for Optional TYPE Operand

```
Path yy is a parallel block-multiplexer channel
Path yy is a parallel byte-multiplexer channel
Path yy is an ESCON unswitched channel
Path yy is an ESCON switched channel
Path yy is an ESCON channel
      (switched or unswitched has not been determined)
Path yy is an ESCON channel to a parallel block-multiplexer converter
Path yy is a native-interface channel
Path yy is an ESCON CTC unswitched channel
Path yy is an ESCON CTC switched channel
Path yy is an ESCON CTC channel
      (switched or unswitched has not been determined)
Path yy is a coupling-facility-sender (CFS) channel
Path yy is a coupling-facility-receiver (CFR) channel
Path yy is an ESCON channel to a parallel byte-multiplexer converter
Path yy is an Open Systems Adapter (OSA) channel
Path yy is an Open Systems Adapter-Express (OSE) channel
Path yy is an Open Systems Adapter Direct Express (OSD) channel
Path yy is an Open Systems Adapter 3270-Console (OSC) channel
Path yy is an Open Systems Adapter NCP (OSN) channel
Path yy is an internal-system-device (ISD) channel
Path yy is a cluster-bus-sender (CBS) channel
Path yy is a cluster-bus-receiver (CBR) channel
Path yy is an internal-coupling-sender (ICS) channel
Path yy is an internal-coupling-receiver (ICR) channel
Path yy is a FICON unswitched channel
Path yy is a FICON switched channel
Path yy is a FICON channel to an ESCON Director FICON Bridge card
Path yy is a FICON channel
      (switched or unswitched has not been determined)
Path yy is a direct-system-device (DSD) channel
Path yy is an emulated-I/O (EIO) channel
Path yy is a cluster-bus-peer (CBP) channel
Path yy is a coupling-facility-peer (CFP) channel
Path yy is an internal-coupling-peer (ICP) channel
Path yy is an internal-queued-direct-communications (IQDC) channel
Path yy is an open-fibre-channel-protocol (FCP) channel
Path yy is a coupling over InfiniBand (CIB) channel
Path yy is an Open Systems Adapter Direct Express IEDN (OSX) channel
Path yy is an Open Systems Adapter Direct Express INMN (OSM) channel
Path yy is an unknown or unrecognized channel type - xx
```

**yy**

is the logical channel path.

**xx**

is the CHPID descriptor value found.

## Usage Notes

The following commands are used to find the virtual CHPID number for a virtual device. In this example, the device number is 191, the subchannel ID is 5, and the CHPID numbers are 60-67.

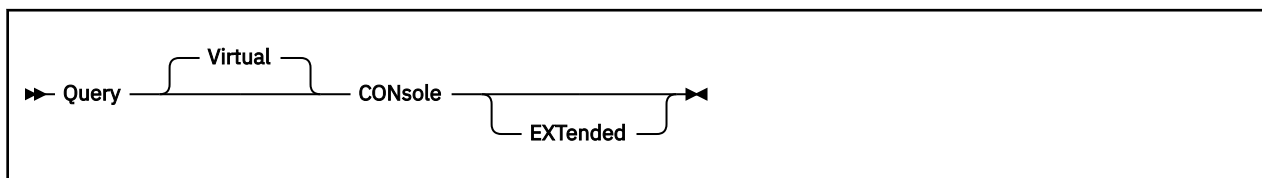
```
QUERY VIRTUAL 191
DASD 0191 3390 USP756 R/W          25 CYL ON DASD 5AA0 SUBCHANNEL = 0005
DISPLAY SCHIB 5
SCHIB DEV INT-PARM ISC FLG LP PNO LPU PI MBI PO PA CHPID0-3 CHPID4-7
0005 0191 00000000 0 83 FF 00 00 FF ---- FF FF 60616263 64656667
```

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid

- HCP846E Invalid channel path identifier.

## QUERY VIRTUAL CONSOLE



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL CONSOLE to display status of the console device accessible to your virtual machine.

### Responses

Response 1:

```

CONS vdev {ON [netname.]luname} {TERM } {STOP } [VSM userid]
           {ON GRAF rdev}      {NOTERM} {START} [HOST userid [FROM ipadr]]
           {ON LDEV ldev}
           {ON LINE rdev}
           {ON SYSC
           }
           {DISCONNECTED}

vdev CL c {CONT } {HOLD } {CPY *nnn} {READY }
          {NOCONT} {NOHOLD} {COPY nnn} {NOTREADY} FORM userform

vdev TO userid {RDR} DIST distcode FLASHC ccc DEST dest
              {PRT}
              {PUN}

vdev FLASH name CHAR char [char[char[char]]] MDFY cmod n FCB fcb LPP nnn

vdev {3215} {EOF } {OPEN spoolid} {KEEP } {MSG }
      {3270} {NOEOF} {CLOSED } {NOKEEP} {NOMSG}

{NAME fname ftype }
{NONAME }

[vdev SUBCHANNEL = vsub ]
    
```

#### **vdev**

is the virtual machine console.

#### **[netname.]luname**

is an SNA/CCS terminal (controlled by a VTAM service machine) that is the virtual console of this virtual machine. It is the name from the VTAM definition of the terminal as an SNA logical unit. The *netname.* is a network qualifier followed by a period delimiter, and is only present when the the EXTended option is specified and the VSM supports network qualified names.

#### **rdev**

is the real or logical device number associated with the virtual console if the console is not disconnected.

#### **ON GRAF rdev**

#### **ON LINE rdev**

#### **ON LDEV ldev**

is the real or logical device number of the virtual console if the console is not disconnected.

#### **ON SYSC**

identifies the system console as being associated with the virtual console.

**DISCONNECTED**

indicates that the console is disconnected.

**TERM****NOTERM**

indicates whether terminal output is or is not produced when console spooling is in effect.

**START****STOP**

indicates whether console spooling has been started or stopped for the device.

**VSM *userid***

provides the user ID of the VTAM service machine managing the device.

**HOST *userid***

provides the user identifier of the virtual machine managing the logical device.

**FROM *ipadr***

provides the IP address of the TCP/IP Telnet client associated with the logical device.

An IPv4 address is displayed in dotted-decimal format.

**Example:** 9 . 130 . 44 . 144

An IPv6 address is displayed in compressed format (":" designates multiple 16-bit groups of zeros).

**Example:** The following is an IPv6 address:

```
F802:0000:0000:0000:0000:0A00:0001
```

This address is displayed in compressed format as:

```
F802::A00:1
```

**CL *c***

identifies the class of spool file created on this device.

**CONT****NOCONT**

controls the continuous spooling option. For output files, CONT causes all CLOSE requests to be ignored unless specified with the EOF override option.

**HOLD****NOHOLD**

controls the HOLD status. For output devices, HOLD prevents files from being punched or printed on a real device by putting a USER HOLD status on the file. If you specify NOHOLD, you override the spool device HOLD option and prevent the file from being placed in HOLD status.

**CPY *\*nnn*****COPY *nnn***

specifies the number of file copies you want printed or punched when the file is produced. When the optional asterisk (\*) is specified, the 3800 printer copies each page from the spooled output files *nnn* times before going on to the next page. If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

**READY****NOTREADY**

indicates whether the device is in a ready or not-ready status. The status can be changed by the READY and NOTREADY commands.

**FORM *userform***

identifies the form associated with the device and assigned to the spool files as they are created.

**TO *userid***

identifies the virtual machine that receives spool files created on the device.

**RDR****PRT****PUN**

identifies the spool file queue onto which spool files created on this device are placed. Unless changed by the SPOOL or CLOSE command, output normally goes to the PRT queue for a console spool file.

**DIST *distcode***

identifies the distribution code for spool files created on this device.

**FLASHC *ccc***

identifies the flash count for the file.

**DEST *dest***

identifies the destination value to be assigned to each spool file created on the device. If the destination value has not been changed by a SPOOL command, the default is OFF.

**FLASH *name***

identifies the forms overlay frame superimposed on the output pages if the file is printed on the 3800 printer.

**CHAR *char***

identifies the name of the character set or sets used to generate output if the file is printed on the 3800 printer.

**MDFY *cmod***

identifies the copy modification module used to alter output text if the file is printed on the 3800 printer.

***n***

identifies the copy modification character set number.

**FCB *fc***

identifies the forms control buffer used in the vertical formatting of a page if the file is printed on the 3800 printer.

**LPP *nnn***

identifies the lines per page setting for the virtual printer where the console output will be generated. The value can be a decimal number in the range 30 to 255, or 'OFF', which indicates that internal defaults will be used.

**3215****3270**

identifies the virtual console device type. A 3270 device type is only supported if the user is logged on at a real 3270.

**EOF****NOEOF**

EOF closes, or NOEOF leaves open, the spool file on the specified device, regardless of the CONT setting for the device.

**OPEN *spoolid*****CLOSED**

indicates that a spool file (*spoolid*) has been created on the console and that the file is still open (OPEN), or indicates that no spool file is open on the device (CLOSED).

**KEEP****NOKEEP**

controls the KEEP status, which indicates whether the current spool file is to be placed in user HOLD after processing. For output files, the KEEP status is placed on the spool file. After it is transcribed at the real printer or punch, the file is assigned user HOLD status, and is not purged. The file's KEEP status is retained. Specify NOKEEP to override the current KEEP setting for the device so that KEEP is not set on the spool file.



**MSG****NOMSG**

indicates whether you want to be informed about file printing and punching. If you specify MSG, you are informed of file printing and punching. The message is sent to you when the file begins printing or punching on a real device. This message is not sent if you enter SET IMSG OFF. Specify NOMSG to reset the MSG option.

**NAME *fname ftype*****NONAME**

identifies the file name and file type, if any, to be assigned to spool files created on the device.

***vsub***

identifies the virtual subchannel number.

**Messages**

- HCP003E Invalid option - *option*
- HCP263E Too many operands were supplied for this command.
- HCP1003E {*CP module name/rdev/ldev*} {is|was} not found.

## QUERY VIRTUAL CPUS



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL CPUS to display a list of CPUs currently defined for your virtual machine.

### Responses

Response 1:

```
CPU nn ID FFssssssmmmm0000 [(BASE)|(ADJUNCT)] status type affinity
```

is displayed for each processor defined in the virtual machine configuration.

#### **nn**

identifies the processor address in hexadecimal, as established by DEFINE CPU.

#### **FFssssssmmmm0000**

identifies the processor associated with the virtual machine.

#### **FF**

identifies the version code. These two digits are forced to X'FF' to indicate that the virtual machine is running under z/VM.

#### **ssssss**

identifies the serial number of the processor as established by SET CPUID or by a CPU directory entry.

#### **mmmm**

identifies the model number of the processor.

#### **0000**

identifies the machine check extended logout length. This field is forced to X'0000'. For a virtual processor using the SET command, these values are the same as for the real processor.

#### **BASE**

indicates that this is the base processor in a virtual MP configuration.

#### **ADJUNCT**

indicates that this is the virtual processor of an adjunct configuration.

#### **status**

indicates the status of the virtual processor. Status can be as follows:

##### **STOPPED**

indicates the virtual processor is in the hard-stopped state.

##### **CHECK-STOPPED**

indicates an error occurred on the virtual processor.

If no status is shown, the processor is running or in the soft-stopped state.

#### **type**

indicates the type of the virtual processor. Type can be as follows:

**CP**

indicates the virtual processor is a central processor.

**IFL**

indicates the virtual processor is an IBM Integrated Facility for Linux. The virtual CPU can run on a real IFL or it can be simulated.

**ZIIP**

indicates the virtual processor is an IBM z Integrated Information Processor. The virtual CPU can run on a real zIIP or it can be simulated.

**ICF**

indicates the Internal Coupling Facility. The virtual CPU can run on a real ICF or it can be simulated.

***affinity***

indicates the affinity of the virtual processor. Affinity can be as follows:

**CPUAFFIN ON**

indicates CPU affinity has been set on for this user with the SET CPUAFFINITY command and this virtual CPU is dispatched on a real CPU of the same type.

**CPUAFFIN SUPP**

indicates CPU affinity is suppressed. Although CPU affinity has been set on for this user with the SET CPUAFFINITY command, there is currently no matching real processor on which to run.

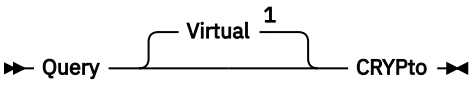
**CPUAFFIN OFF**

indicates CPU affinity has not been set on for this user. When OFF is in effect the virtual CPU is dispatched on only real CPUs of primary type.

**Messages**

- HCP263E Too many operands were supplied for this command.

## QUERY VIRTUAL CRYPTO



Notes:

<sup>1</sup> This operand is required for privilege class G users who also have privilege class A, B, C, or E.

### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL CRYPTO to display the status of the virtual crypto resources.

### Responses

Response 1:

The following response is displayed when one or more crypto resources are attached to this virtual machine.

```
AP apnum CEXtm Domain domnum {shared|dedicated} config_state
```

#### **apnum**

is a three-digit virtual crypto adapter number (in decimal). If the resource is shared, then 001 is assigned to the virtual adapter number. If the resource is dedicated, then the actual hardware adapter number of the resource is assigned.

#### **CEXtm**

is the virtual crypto adapter, where *t* indicates the adapter type and *m* indicates the adapter mode. For dedicated resources, the value indicates the physical resource. For shared resources, the value indicates the lowest type of crypto express adapter that is available in the shared pools of all systems in the user's relocation domain. Shared crypto modes are accelerator mode or CCA coprocessor mode.

For more information, see [Crypto Planning and Management in z/VM: CP Planning and Administration](#).

The CEXtm value can be one of the following:

#### **CEX3C**

Crypto Express3 configured in CCA coprocessor mode

#### **CEX3A**

Crypto Express3 configured in accelerator mode

#### **CEX4C**

Crypto Express4 configured in CCA coprocessor mode

#### **CEX4A**

Crypto Express4 configured in accelerator mode

#### **CEX4P**

Crypto Express4 configured in EP11 coprocessor mode

#### **CEX5C**

Crypto Express5 configured in CCA coprocessor mode

**CEX5A**

Crypto Express5 configured in accelerator mode

**CEX5P**

Crypto Express5 configured in EP11 coprocessor mode

**CEX6C**

Crypto Express6 configured in CCA coprocessor mode

**CEX6A**

Crypto Express6 configured in accelerator mode

**CEX6P**

Crypto Express6 configured in EP11 coprocessor mode

**CEX7C**

Crypto Express7 configured in CCA coprocessor mode

**CEX7A**

Crypto Express7 configured in accelerator mode

**CEX7P**

Crypto Express7 configured in EP11 coprocessor mode

**CEX8C**

Crypto Express8 configured in CCA coprocessor mode

**CEX8A**

Crypto Express8 configured in accelerator mode

**CEX8P**

Crypto Express8 configured in EP11 coprocessor mode

**NONE**

There are no crypto resources attached to the shared pool.

**UNKN**

z/VM is unable to determine the type of crypto adapter.

***domnum***

is a three-digit virtual domain number (in decimal). If the resource is shared, then 001 is assigned to the virtual domain number. If the resource is dedicated, then the actual hardware adapter number of the resource is assigned.

***config\_state***

is CP's logical view of the resource state, as controlled by VARY CRYPTO.

For an APVIRT virtual machine, the configuration state is reported for the pool of shared resources. The shared pool could have one of the following configuration states:

- online - At least one shared crypto resource is online and operational.
- offline - All shared crypto resources are offline.
- unavailable - No crypto resources are attached to SYSTEM.
- error - All online shared resources are in error recovery.

For an APDED virtual machine, the configuration state is reported for each crypto resource dedicated to the virtual machine. A dedicated crypto resource could have one of the following configuration states:

- online - Dedicated crypto resource is online.
- offline - Dedicated crypto resource is offline and not available for use.

**Note:**

1. An adapter must be configured in CCA coprocessor mode or accelerator mode in order to be attached to the system for shared use.
2. An adapter that is configured in CCA coprocessor mode, accelerator mode or EP11 coprocessor can be attached to a virtual machine for dedicated use.

## QUERY VIRTUAL CRYPTO

3. For information on planning and managing crypto resources on a z/VM system, see [Crypto Planning and Management](#) in *z/VM: CP Planning and Administration*.

Response 2:

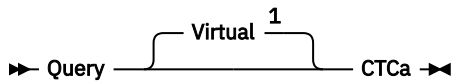
The following response is displayed when no crypto resources are available to the guest virtual machine.

```
No AP crypto domains are available
```

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid

## QUERY VIRTUAL CTCA



Notes:

<sup>1</sup> This operand is required for privilege class G users who also have privilege class B.

### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL CTCA to display status of the channel-to-channel adapters accessible to your virtual machine.

### Responses

Response 1:

```
CTCA  vdev1 subclass {NOT READY [FOR userid2] } [SUBCHANNEL = vsub]
                   {COUPLED TO  userid vdev2 }
```

is displayed for each virtual channel-to-channel adapter in your virtual machine system.

Response 2:

```
CTCA  vdev1  ON DEV rdev [SUBCHANNEL = vsub]
```

is returned when your CTCA is a real device dedicated to your virtual machine.

#### **vdev1**

identifies the virtual device number of the CTCA.

#### **subclass**

identifies the subclass of the virtual CTCA (3088, SCTC, BCTC, CNC, or FCTC).

#### **userid**

identifies the virtual machine to which your CTCA is connected.

#### **userid2**

identifies the user authorized to couple to this device. This is configured on the SPECIAL directory statement in the user directory.

#### **vdev2**

identifies the virtual device number of the CTCA connected by the virtual machine identified as *userid*.

#### **rdev**

identifies the real device number of your CTCA.

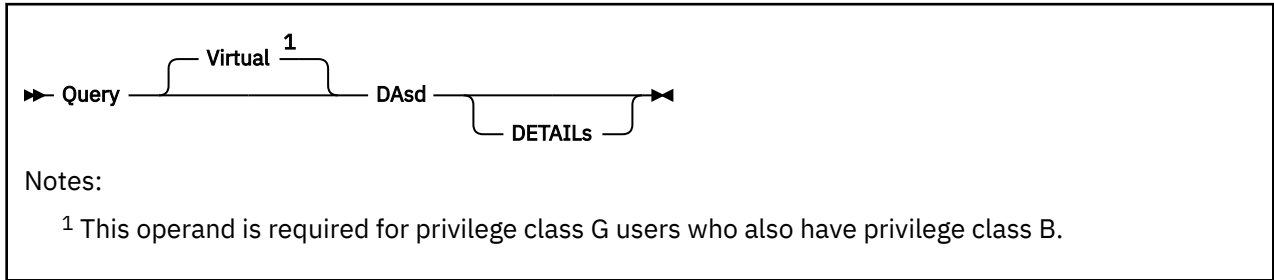
#### **vsub**

identifies the virtual subchannel number.

### Messages

- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP263E Too many operands were supplied for this command.

## QUERY VIRTUAL DASD



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL DASD to display the status of all DASDs accessible to your virtual machine, including temporary disks (T-disks) and virtual disks in storage.

### Operands

#### DETAILS

displays additional information for DASD as well as the usual response, if applicable.

### Usage Notes

1. The DETAILS operand is meaningful only for dedicated DASDs or minidisk DASDs on a cache subsystem. The response for other virtual DASD remains the same.

### Responses

Response 1:

```
DASD vdev type {valid} {R/W} {nnnnnnn CYL} ON DASD rdev [SUBCHANNEL=vsub]
                {(TEMP)} {R/O} {nnnnnnn BLK}
```

displays for each minidisk or temporary disk in your virtual machine.

#### *vdev*

is the virtual device number.

#### *type*

is an IBM direct access device type.

#### *valid*

#### (TEMP)

#### (VDSK)

identifies the volume label of the real device on which your virtual disk resides. (TEMP) indicates that your virtual DASD has been defined from system temporary disk space (a T-disk). (VDSK) indicates that your virtual DASD has been simulated in system storage (a virtual disk in storage).

#### R/W

#### R/O

identifies the type of access you have to the disk. R/O is read-only access; R/W is read and write access.

#### *nnnnnnn* CYL

identifies the number of cylinders on the virtual CKD or ECKD DASD device.



**nnnnnnn BLK**

identifies the number of blocks on the virtual FBA DASD device.

**rdev**

**VDSK**

identifies the real device number of the volume containing the virtual device. VDSK is displayed for a virtual disk in storage because it is not mapped to a real device.

**vsub**

identifies the virtual subchannel number.

Response 2:

```
DASD vdev type (VDSK) {R/W} nnnnnnn BLK ON DASD VDSK [SUBCHANNEL=vsub]
                    {R/O}
```

is displayed for each virtual disk in storage attached to your virtual machine. For an explanation of the response contents, see **Response 1**.

Response 3:

```
DASD vdev ON DASD rdev {R/W} valid [SUBCHANNEL = vsub]
                       {R/O}
```

is displayed for each real device dedicated to your virtual machine. For an explanation of the response contents, see **Response 1**.

Response 4:

The following line is displayed in addition to **Response 1** or **Response 3** when the DETAILS operand is specified in the command:

```
[SYSCTL | DEVCTL | NOCTL | CACHE | NOCACHE | STABLE | EXCLUSIVE]
[PRIMARY DUPLEX | SECONDARY DUPLEX][REAL-MPLF | SIMULATED-MPLF]
[PAVBASE (nn) | PAVALIAS (vdev/rdev) | HYPERPAVBASE (poolnum) |
HYPERPAVALIAS (rdev,poolnum)]
[ALIASES: vdev/rdev vdev/rdev vdev/rdev vdev/rdev][SEF]
```

**SYSCTL**

specifies that CCWs are accepted which have a direct effect on global subsystem resources and functions, and not just those related to the device.

**DEVCTL**

specifies that CCWs are accepted that have an effect on resources and functions directly related to the device.

**NOCTL**

No CCWs are accepted that can exercise control of subsystem resources and functions, regardless of whether they are directly related to the device.

**CACHE**

means that the minidisk has access to control unit cache storage.

**NOCACHE**

means that CP forces I/O to the minidisk to bypass storage control cache.

**STABLE**

means that the disk was acquired with either an SR, SW, or SM access mode, preventing all other users from getting write access to this disk.

**EXCLUSIVE**

means that the disk was acquired with either an ER or EW link mode, preventing all other users from getting any access to this disk.

**PRIMARY DUPLEX**

means that the disk is on the primary volume of a duplex pair.

**SECONDARY DUPLEX**

means that the disk is on the secondary volume of a duplex pair.

## QUERY VIRTUAL DASD

### REAL-MPLF

means that the device is enabled to use the multi-path lock facility on the real control unit.

### SIMULATED-MPLF

means that the device is enabled to use VM's simulation of the multi-path lock facility.

### PAVBASE (*nn*)

indicates that the device is a base Parallel Access Volume. *nn* indicates the number of alias Parallel Access Volumes that are associated with this base.

### PAVALIAS (*vdev/rdev*)

indicates that the device is an alias Parallel Access Volume. *vdev* indicates the device number of the associated virtual base Parallel Access Volume and *rdev* indicates the device number of the assigned real alias Parallel Access Volume.

### HYPERPAVBASE (*poolnum*)

indicates that the device is a base HyperParallel Access Volume (HyperPAV). *poolnum* indicates the associated pool for this HyperPAV base device.

### HYPERPAVALIAS (*rdev,poolnum*)

indicates that the device is an alias HyperParallel Access Volume (HyperPAV). *rdev* indicates the device number of the assigned real HyperPAV alias device. *poolnum* indicates the associated pool for this HyperPAV alias device.

### ALIASES: *vdev/rdev vdev/rdev vdev/rdev vdev/rdev*

indicates the list of alias Parallel Access Volumes that are associated with the base Parallel Access Volume (PAVBASE (*nn*)). *vdev* indicates the device number of the virtual alias Parallel Access Volume and *rdev* indicates the device number of the assigned real alias Parallel Access Volume. The number of *vdev/rdev* pairs that are displayed equals the (*nn*) number after the PAVBASE parameter. A maximum of four *vdev/rdev* pairs will be listed per response line.

### SEF

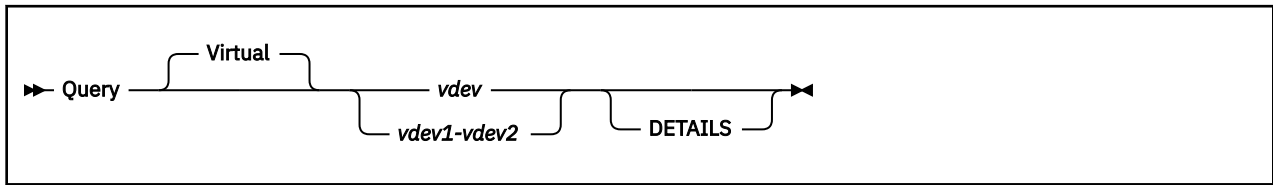
specifies that the underlying real device is a space-efficient volume.

```
q v dasd details
DASD 019E 3390 PACK03 R/O      355 CYL ON DASD  5C81 SUBCHANNEL = 0011
CACHE
DASD 0200 3390 PACK01 R/W      100 CYL ON DASD  5100 SUBCHANNEL = 0002
CACHE SEF
```

## Messages

- HCP003E Invalid option - *option*.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command.

## QUERY (Virtual Device)



### Authorization

Privilege Class: G

### Purpose

Use QUERY (virtual device) to display the status of devices accessible to your virtual machine. These devices may be located at a specific virtual device number or located in a range of virtual device numbers.

### Operands

**vdev**

**vdev1-vdev2**

identifies the device or range of virtual device numbers for devices whose status is to be displayed.

**DETAILS**

displays additional information for the specified virtual device or range of devices.

### Usage Notes

1. The DETAILS option is meaningful only for dedicated DASDs, or minidisk DASDs, on a cache subsystem, or for tapes. The responses for devices other than DASD, if specified, remain the same.

### Responses

For devices not listed below, see the QUERY (individual device type) command description for their response formats.

Response 1:

Responses you receive depend on the type of device located at the device number you specify. These responses are included in the QUERY VIRTUAL commands for those devices.

You receive this response if the device is one of the following:

- 3745 Communications Controller
- A Mass Storage Controller
- A virtual 270x line
- A message device
- A tape device
- A miscellaneous device (one not explicitly described in the QUERY VIRTUAL family of commands).

```
type vdev {ON DEV rdev          } [SUBCHANNEL = vsub] [encap]
          {ON NIC vdev AS UNIT unit}
```

**type**

identifies the virtual device type; it can be any of the following:

## QUERY (Virtual Device)

<u>Type</u>	<u>Meaning</u>
CTLR	3745 Communications Controller
MSC	Mass Storage Controller
LINE	Virtual 270x line
MSGD	Message Device
OSA	Open Systems Adapter
FCP	SCSI Device Adapter
TAPE	Tape Device
DEV	Any miscellaneous device

### **vdev**

identifies the virtual device number.

### **ON DEV rdev**

identifies the real device number.

### **ON NIC vdev AS UNIT unit**

identifies the base device (for a virtual network adapter) and the relative device (starting with 0).

### **vsub**

identifies the virtual subchannel number.

### **enccap**

If the device is capable of hardware encryption, the string "ENCRYPTION CAPABLE" will be included. Otherwise, this field will be blank.

Response 2:

```
[SYSCTL | DEVCTL | NOCTL | CACHE | NOCACHE | STABLE | EXCLUSIVE]
[PRIMARY DUPLEX | SECONDARY DUPLEX]
[REAL-MPLF | SIMULATED-MPLF]
[PAVBASE (nn) | PAVALIAS (vdev/rdev) | HYPERPAVBASE (poolnum) |
HYPERPAVALIAS (rdev,poolnum)]
[ALIASES: vdev/rdev vdev/rdev vdev/rdev vdev/rdev]
[ACTIVE KEY LABEL(S): keys]
[ATTACHED KEY LABEL(S): keys]
[INACTIVE KEY LABEL(S): keys]
[Position: pos BLK Relative position: relpos%]
```

is the response displayed, in addition to the normal QUERY VIRTUAL DEVICE response, if the DETAILS option is specified.

### **SYSCTL**

specifies that CCWs are accepted that have a direct effect on global subsystem resources and functions and not just those related to the device.

### **DEVCTL**

specifies that CCWs are accepted that have an effect on resources and functions directly related to the device.

### **NOCTL**

No CCWs are accepted that can exercise control of subsystem resources and functions, regardless of whether they are directly related to the device.

### **CACHE**

indicates that the minidisk has access to storage control cache.

### **NOCACHE**

indicates that CP forces I/O to the minidisk to bypass storage control cache.

### **STABLE**

indicates that the disk was acquired with either an SR, SW, or SM access mode, preventing all other users from getting write access to this disk.

**EXCLUSIVE**

indicates that the disk was acquired with either an ER or EW link mode, preventing all other users from getting any access to this disk.

**PRIMARY DUPLEX**

means that the disk is on the primary volume of a duplex pair.

**SECONDARY DUPLEX**

means that the disk is on the secondary volume of a duplex pair.

**REAL-MPLF**

means that the device is enabled to use the multi-path lock facility on the real control unit.

**SIMULATED-MPLF**

means that the device is enabled to use VM's simulation of the multi-path lock facility.

**PAVBASE (nn)**

indicates that the device is a base Parallel Access Volume. *nn* indicates the number of alias Parallel Access Volumes that are associated with this base.

**PAVALIAS (vdev/rdev)**

indicates that the device is an alias Parallel Access Volume. *vdev* indicates the device number of the associated virtual base Parallel Access Volume and *rdev* indicates the device number of the assigned real alias Parallel Access Volume.

**HYPERPAVBASE (poolnum)**

indicates that the device is a base HyperParallel Access Volume (HyperPAV). *poolnum* indicates the associated pool for this HyperPAV base device.

**HYPERPAVALIAS (rdev,poolnum)**

indicates that the device is an alias HyperParallel Access Volume (HyperPAV). *rdev* indicates the device number of the assigned real HyperPAV alias device. *poolnum* indicates the associated pool for this HyperPAV alias device.

**ALIASES: vdev/rdev vdev/rdev vdev/rdev vdev/rdev**

indicates the list of alias Parallel Access Volumes that are associated with the base Parallel Access Volume (PAVBASE (*nn*)). *vdev* indicates the device number of the virtual alias Parallel Access Volume and *rdev* indicates the device number of the assigned real alias Parallel Access Volume. The number of *vdev/rdev* pairs that are displayed equals the (*nn*) number after the PAVBASE parameter. A maximum of four *vdev/rdev* pairs will be listed per response line.

**ACTIVE KEY LABEL(S): keys**

indicates the active key labels that are currently associated with the attached device. This is usually through either an ATTACH command with the KEY option, or by an unmodified ATTACH that follows a SET RDEV command with the FEATURE KEY option, but this could also be the case when a tape is mounted that was previously encrypted with different key labels or had been rekeyed. If "keys" indicates "DEFAULT", the external key manager's default keys will be used. Otherwise, will indicate the encoding mechanism ("L" for Label, "H" for Hash of Label) and the associated key label that was specified.

**ATTACHED KEY LABEL(S): keys**

indicates the key labels that were specified through the ATTACH command. This is only displayed if it differs from what is currently in use on the tape cartridge and displayed under the "active" heading.

**INACTIVE KEY LABEL(S): keys**

indicates the key labels that are associated with the device as a result of a SET RDEV command with the FEATURE KEY option. If "keys" indicates "DEFAULT", the external key manager's default keys will be used. Otherwise, will indicate the encoding mechanism ("L" for Label, "H" for Hash of Label) and the associated key label that was specified.

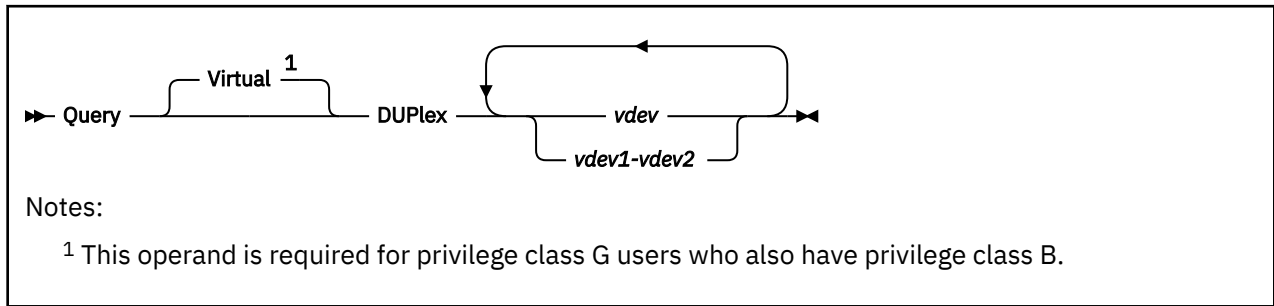
**Position: pos BLK Relative position: relpos%**

indicates the current tape block position and relative position. Note that this line is shown only for the 3592 Model E07 and higher.

### Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command.
- HCP1003E {*CP module name/rdev/ldev*} {*is|was*} not found.
- HCP6005E Option *option* is not supported by z/VM.

## QUERY VIRTUAL DUPLEX



### Authorization

Privilege Class: G

### Purpose

Use the QUERY VIRTUAL DUPLEX command to display the status of virtual duplex pairs attached to 3990-3 and 3990-6 storage subsystems.

### Operands

**vdev**

**vdev1-vdev2**

is the virtual device number, a list of virtual device numbers, or a range of virtual device numbers of the primary or secondary volumes in the established duplex pairs.

### Usage Notes

1. The device number of either the primary or secondary volume in a duplex pair may be specified on this command.
2. If both the primary and secondary of a duplex pair are included in the device list or device range, two identical responses will be displayed.

### Responses

Response 1:

```
Duplex pair (pvdev,svdev) - active
```

is displayed for each duplex pair that is not in the *suspended duplex* state and for which synchronization is not in progress. If either the primary or secondary device is not a virtual device for the user, hyphens (----) will appear in the corresponding *pvdev* or *svdev* position.

**pvdev**

is the virtual device number of the primary volume in the duplex pair.

**svdev**

is the virtual device number of the secondary volume in the duplex pair.

Response 2:

```
Duplex pair (pvdev,svdev) - copy operation is in progress.
```

is displayed for each duplex pair that is in the process of being synchronized. If either the primary or secondary device is not a virtual device for the user, hyphens (----) will appear in the corresponding *pvdev* or *svdev* position.

***pvdev***

is the virtual device number of the primary volume in the duplex pair.

***svdev***

is the virtual device number of the secondary volume in the duplex pair.

Response 3:

```
Duplex pair (pvdev,svdev) - suspended
```

is displayed for each duplex pair that is in the *suspended duplex* state. Either the primary or the secondary volume has been suspended. If either the primary or secondary device is not a virtual device for the user, hyphens (----) will appear in the corresponding *pvdev* or *svdev* position.

***pvdev***

is the virtual device number of the primary volume in the duplex pair.

***svdev***

is the virtual device number of the secondary volume in the duplex pair.

Response 4:

```
vdev The associated {primary|secondary}  
volume in the duplex pair is not online to VM.
```

is displayed when the virtual device number of the primary or secondary volume is specified on the command and the associated real device number is not in the CP I/O configuration.

***vdev***

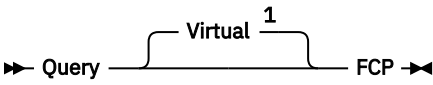
is the virtual device number of the primary or secondary volume in the duplex pair.

### Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP009E Invalid range - *range*
- HCP022E A virtual device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP319E Error processing subsystem command - *cmd* for DASD *vdev*
- HCP332E Invalid control unit type - *vdev*
- HCP1110E Device *rdev/vdev* is not in a duplex pair.



## QUERY VIRTUAL FCP



Notes:

<sup>1</sup> This operand is required for privilege class G users who also have privilege class B.

### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL FCP to display status of the Fibre Channel Protocol (FCP) adapter devices accessible to your virtual machine.

### Responses

For each FCP device dedicated to your virtual machine configuration, the following is displayed:

```
FCP vdev ON DEV rdev UNIT unit CHPID chp SUBCHANNEL = vsub [TOKEN = token]
      vdev DEVTYP devtyp CHPID chp chptyp
      vdev [QDIO-ELIGIBLE|QDIO PENDING ACTIVE|QDIO ACTIVE|QIOASSIST status] [(QEBSM)]
      vdev [DATA ROUTER ELIGIBLE|DATA ROUTER ACTIVE]
```

If the subchannel for the FCP device is in either QDIO PENDING ACTIVE or QDIO ACTIVE state, the following is displayed for each input and output data queue that was established by a program running in a V=V virtual machine only:

```
vdev type sync priority IOCNT = nnnnnnnn ADP = nnn PROG = nnn UNAVAIL = nnn
vdev BYTES = nnnnnnnnnnnnnnnnnn
```

The terms shown in the above Results have the following meanings:

***vdev***

identifies the virtual device number of the FCP device.

***rdev***

identifies the real device number of the FCP device.

**CHPID *chp***

identifies the real CHPID number of the FCP device.

**SUBCHANNEL = *vsub***

identifies the virtual subchannel number associated with the attached FCP device.

**TOKEN = *token***

identifies the subchannel token assigned to the QDIO connection when it is managed by QDIO enhanced buffer state management (QEBSM).

**DEVTYP *devtyp***

identifies the real device type of the FCP device.

**CHPID *chp chptyp***

identifies the real CHPID number and CHPID type of the FCP device.

**QDIO-ELIGIBLE**

indicates that the device is eligible to use the QDIO Hardware Facility for direct data transfer.

### **QDIO PENDING ACTIVE**

indicates that a program running in the virtual machine has established the QDIO data queues for transferring data between the program and the specified FCP device. Although the data queues are created, they must be activated by the program running in the virtual machine before data transfer can commence.

### **QDIO ACTIVE**

indicates that a program running in the virtual machine has activated the QDIO data queues. Direct data transfer between the specified FCP device and the program may now occur.

### **QIOASSIST status**

The status is one of the following:

- **QIOASSIST NOT AVAILABLE** – indicates that the processor does not support the queue-I/O assist (QDIO performance assist for V=V guests), or z/VM is running in an environment where the queue-I/O assist is not available. See the SET QIOAssist command for a list of requirements to use the queue-I/O assist.
- **QIOASSIST- ELIGIBLE** – indicates the device is eligible to use the queue-I/O assist (QDIO performance assist for V=V guests), when the virtual machine begins using the device with no traces in effect.
- **QIOASSIST DISABLED** – indicates the device is not permitted to use the queue-I/O assist (QDIO performance assist for V=V guests). The device must have been attached to the virtual machine with the NOQIOAssist option either on the CP ATTACH command or the system-directory DEDICATE statement.
- **QIOASSIST ACTIVE** – indicates the queue-I/O assist (QDIO performance assist for V=V guests) is active and in-use.

### **QEBSM**

indicates that the device connection is managed using enhanced QDIO buffer state management protocol. When QEBSM manages a QDIO connection, a QDIO I/O request can be initiated without z/VM involvement.

### **DATA ROUTER ELIGIBLE**

indicates that the device is eligible to use the Data Router facility to directly transfer data from guest memory to the FCP device.

### **DATA ROUTER ACTIVE**

indicates that a program running on the virtual machine requested that the Data Router facility be used when transferring data between guest memory and the FCP device.

### **type**

indicates the type of QDIO data queue:

#### **INP**

Input

#### **OUT**

Output

### **sync**

indicates whether the program and CP versions of the QDIO data queue are currently synchronized:

+

Input

-

Output

### **priority**

indicates the priority of the QDIO data queue. All input and output queues are listed according to their queue priority, with the lowest number queues having the highest priority.

**IOCNT = *nnnnnnnn***

specifies the number of data transfers that have completed since the QDIO data queue was last activated by the program, in decimal. CP increments the IOCNT each time a QDIO data buffer changes from an adapter-owned state to a program-owned state.

**BYTES = *nnnnnnnnnnnnnnnn***

specifies the number of bytes transferred on that queue since device activation, in hexadecimal.

**ADP = *nnn***

specifies the number of QDIO data buffers currently pending or available for a direct data transfer. This is the number of QDIO data buffers currently owned by the adapter.

**PROG = *nnn***

specifies the number of QDIO data buffers currently owned by the program running in the virtual machine that have storage blocks allocated.

**UNAVAIL = *nnn***

specifies the number of QDIO data buffers currently owned by the program running in the virtual machine that have no storage blocks allocated.

**Examples**

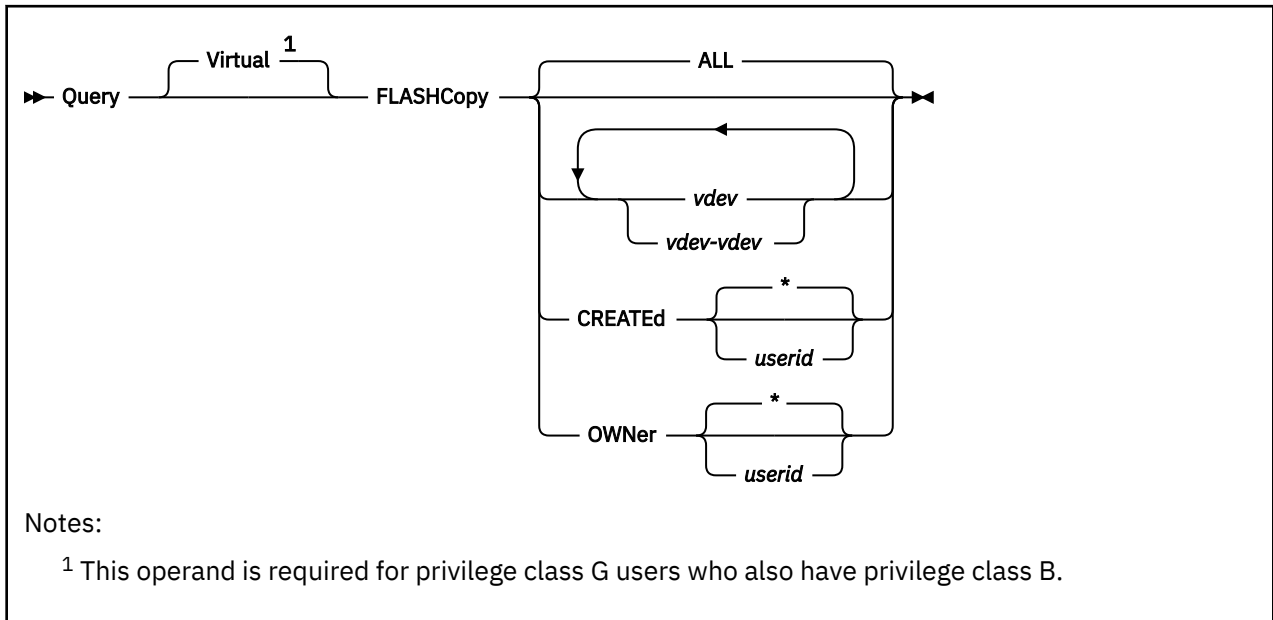
This example displays three virtual FCP devices within the virtual machines' I/O configuration:

```
cp query virtual fcp
FCP 0500 ON FCP 0500 CHPID 80 SUBCHANNEL = 000D
    0500 DEVTYPE FCP          CHPID 80 FCP
    0500 QDIO-ELIGIBLE
FCP 0501 ON FCP 0500 CHPID 80 SUBCHANNEL = 000E
    0501 DEVTYPE FCP          CHPID 80 FCP
    0501 QDIO-ELIGIBLE
FCP 0502 ON DEV 0500 CHPID 80 SUBCHANNEL = 000F
    0502 DEVTYPE FCP          CHPID 80 FCP
    0502 QDIO-ELIGIBLE
```

**Messages**

- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command

## QUERY VIRTUAL FLASHCOPY



### Authorization

Privilege Class: G

### Purpose

This command allows interrogation of persistent FlashCopy relationships in the Persistent FlashCopy Relationship Table. Each FlashCopy relationship is identified with a unique sequence number that is derived from the TOD clock when the relationship is established.

The Persistent FlashCopy Relationship Table is maintained to preserve additional information regarding relationships. Note that this table is NOT maintained over system IPLs and is intended to be used as supplemental information regarding FlashCopy relationships.

### Operands

#### **vdev**

defines a virtual device in your current virtual configuration that is to be interrogated.

#### **ALL**

specifies that all virtual devices are to be interrogated.

#### **CREATED \***

#### **CREATED *userid***

displays additional information regarding persistent FlashCopy relationships that were established by the current user during the current system IPL session. When specifying a *userid*, it must be the user issuing this command.

#### **OWNer \***

#### **OWNer *userid***

displays additional information regarding persistent FlashCopy relationships that were established on minidisks owned by the current user during the current system IPL session. When specifying a *userid*, it must be the user issuing this command.

## Usage Notes

1. The contents of the Persistent FlashCopy Relationship Table are lost when the system is IPLed.
2. Entries might exist in the list for relationships that have been partially withdrawn by this or another LPAR.
3. Entries will not exist for relationships that have been completely withdrawn.

## Responses

The responses you receive will depend on the device extents, full-pack or minidisk, for the device numbers being returned. The number of active FLASHCOPY relationships is defined within the extent of the virtual device, while the maximum number of FLASHCOPY relationships is a limit for the entire volume on which the virtual device is defined. When the active number of FLASHCOPY relationships approaches the maximum number, defining new FLASHCOPY relationships might be limited due to unavailable resources within the Enterprise Storage Server or by extent conflicts for either the source or target relationships.

```
q v flashcopy F48 F66
VDEVN SSID LSS UA CU SERIAL ACTIVE MAX PPRC
0F48 A760 00 28 2105 0000000FCA76 263 951 Secondary
0F66 B660 01 30 2105 0000000FCB66 0 0
```

where:

Field	Meaning
VDEVN	Virtual device number
SSID	Subsystem identifier
LSS	Logical subsystem number
UA	Unit address
CU	Control unit type
SERIAL	Storage facility serial number
ACTIVE	Number of active FLASHCOPY relationships
MAX	Maximum number of FLASHCOPY relationships allowed
PPRC	Peer-to-Peer Remote Copy status
	Primary - device is a Peer-to-Peer Primary volume
	Secondary - device is a Peer-to-Peer Secondary volume
	Cascading - device is a Peer-to-Peer Cascading volume
	Blank - device is not active for Peer-to-Peer copy

```
q v flashcopy
VDEVN SSID LSS UA CU SERIAL ACTIVE MAX PPRC
0200 0101 01 00 2107 000000004191 1 65534
0210 0101 01 00 2107 000000004191 1 65534
0220 0101 01 00 2107 000000004191 0 65534
0291 010E 0E 01 2107 000000004191 0 50099
0300 0101 01 01 2107 000000004191 1 65534
0310 0101 01 01 2107 000000004191 1 65534
0320 0101 01 01 2107 000000004191 0 65534
0500 010E 0E 01 2107 000000004191 0 50099
2000 0101 01 00 2107 000000004191 0 65534
3000 0101 01 01 2107 000000004191 0 65534
FCE0 0201 01 00 2107 0000000T5871 0 50099
FCE1 0201 01 01 2107 0000000T5871 0 50099
Ready; T=0.01/0.01 15:46:46
```

```
q flashcopy created rws
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
4B145C34 2008-04-30 15:19:55 5100> PACK01 RWS RWS 0200
4B145C34 2008-04-30 15:19:55 >5101 PACK02 RWS RWS 0300
4B14700A 2008-04-30 15:20:00 5100> PACK01 RWS RWS 0210
4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS RWS 0310
Ready; T=0.01/0.01 15:20:53
```

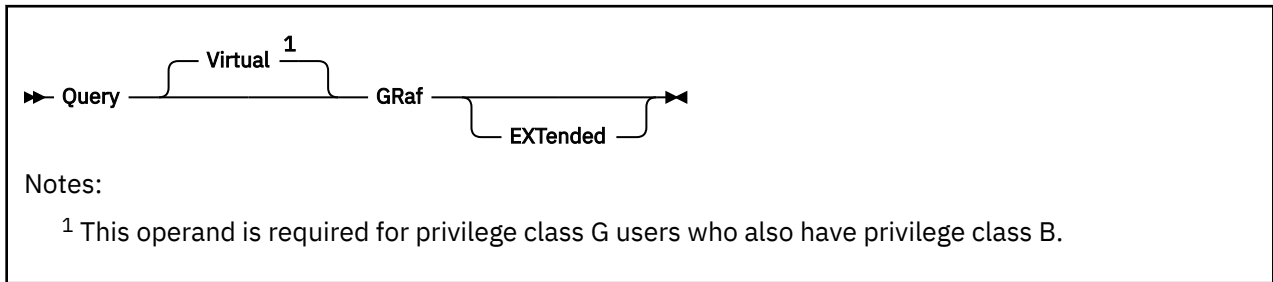
```
q flashcopy owner rws
SEQUENCE ---DATE--- --TIME-- RDEV VOLSER CREATOR OWNER VDEV
4B145C34 2008-04-30 15:19:55 5100> PACK01 RWS RWS 0200
4B145C34 2008-04-30 15:19:55 >5101 PACK02 RWS RWS 0300
4B14700A 2008-04-30 15:20:00 5100> PACK01 RWS RWS 0210
```

4B14700A 2008-04-30 15:20:00 >5101 PACK02 RWS RWS 0310  
Ready; T=0.01/0.01 15:21:53

### Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP021E A device number was not supplied or it is invalid
- HCP022E A virtual device number was not supplied or it is invalid
- HCP026E Operand missing or invalid
- HCP040E Device *vdev* does not exist
- HCP292E A virtual device was not found that supports this function
- HCP332E Invalid control unit type - *rdev|vdev*
- HCP2465I No matches found.

# QUERY VIRTUAL GRAF



## Authorization

Privilege Class: G

## Purpose

Use QUERY VIRTUAL GRAF to show the status of all display devices attached to your virtual machine.

## Operands

### EXTended

specifies that logical unit network qualifiers, or logical device host names and IP addresses, are displayed, if they exist.

An IPv4 address is displayed in dotted-decimal format.

**Example:** 9 . 130 . 44 . 144

An IPv6 address is displayed in compressed format (":" designates multiple 16-bit groups of zeros).

**Example:** The following is an IPv6 address:

```
F802:0000:0000:0000:0000:0A00:0001
```

This address is displayed in compressed format as:

```
F802::A00:1
```

## Responses

Response 1:

You receive the following response for each of your locally attached display devices:

```
GRAF vdev {ON GRAF rdev          } [SUBCHANNEL = vsub]
           {ON LDEV ldev         }
           {ON [netname.]luname VSM userid}
           {NOT DIALED          }
```

### vdev

identifies the virtual display device.

### ON GRAF rdev

### ON LDEV ldev

shows the real (*rdev*) or logical (*ldev*) device that is dialed into, or attached to, this virtual machine.

### ON [netname.]luname

identifies an SNA/CCS terminal (controlled by the VTAM service machine) that is dialed into this virtual machine. It is the name from the VTAM definition of the terminal as an SNA logical unit. The *netname*.

## QUERY VIRTUAL GRAF

is a network qualifier followed by a period delimiter, and is only present when the EXTended option is specified and the VSM supports network qualified names.

### **VSM *userid***

identifies the user ID of the VTAM service machine managing the device.

### **NOT DIALED**

shows the status of a virtual display device that has not been attached using the DIAL command.

### ***vsub***

identifies the virtual subchannel number.

Response 2:

```
CONS vdev {ON GRAF rdev } [SUBCHANNEL = vsub]  
          {ON LDEV ldev }  
          {ON luname VSM userid }  
          {ON [netname.]luname VSM userid}  
          {DISCONNECTED }
```

is returned if your virtual console is conmode 3270.

### ***vdev***

identifies the virtual display device.

### **ON GRAF *rdev***

### **ON LDEV *ldev***

shows the real (*rdev*) or logical (*ldev*) device that is logged onto this virtual machine.

### **ON [*netname.*]*luname***

identifies an SNA/CCS terminal (controlled by the VTAM service machine) that is logged onto this virtual machine. It is the name from the VTAM definition of the terminal as an SNA logical unit. The *netname.* is a network qualifier followed by a period delimiter, and is only present when the EXTended option is specified and the VSM supports network qualified names.

### **VSM *userid***

identifies the user ID of the VTAM service machine managing the device.

### **DISCONNECTED**

indicates that the console is disconnected.

### ***vsub***

identifies the virtual subchannel number.

Response 3:

```
PRT vdev { ON DEV rdev } [SUBCHANNEL = vsub]  
          { ON DEV ldev }
```

is returned for each of your locally attached display 328x printers.

### ***vdev***

identifies the virtual display 328x printer.

### **ON DEV *rdev***

### **ON DEV *ldev***

shows the real (*rdev*) or logical (*ldev*) device number of the display 328x printer attached to this virtual machine.

### ***vsub***

identifies the virtual subchannel number.

## Messages

- HCP003E Invalid option - *option*
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command.



# QUERY VIRTUAL LINES

Notes:

<sup>1</sup> This operand is required for privilege class G users who also have privilege class B.

## Authorization

Privilege Class: G

## Purpose

Use QUERY VIRTUAL LINES to display the status of all of your virtual communications lines.

## Usage Notes

1. The QUERY LINES, QUERY LINKS, and QUERY VIRTUAL LINES commands share the two possible abbreviations "LI" and "LIN". If you are authorized to issue these commands, we recommend that you specify at least 4 character abbreviations of the LINES or LINKS operand to ensure that you invoke the intended command. For example, instead of "Q LI" specify "Q LINE".

## Responses

Response 1:

```
LINE vdev {ON LINE rdev} [SUBCHANNEL = vsub]
          {ENABLED      }
          {DISABLED     }
```

is returned for each of your communications lines.

### **vdev**

identifies the virtual device number of the communication line.

### **ON LINE rdev**

shows the real (*rdev*) device dialed into, or attached to, this virtual machine.

### **ENABLED**

shows the status of a virtual communication line that is enabled.

### **DISABLED**

shows the status of a virtual communication line that is disabled.

### **vsub**

identifies the virtual subchannel number.

Response 2:

```
CONS vdev {ON LINE rdev          } [SUBCHANNEL = vsub]
          {ON GRAF rdev          }
          {ON LDEV ldev          }
          {ON [netname.]luname VSM userid}
          {ON SYSC                }
          {DISCONNECTED          }
```

is returned for your virtual console if your virtual console is conmode 3215.

## QUERY VIRTUAL LINES

### ***vdev***

identifies the virtual device number of the communication line.

### **ON LINE *rdev***

### **ON GRAF *rdev***

### **ON LDEV *ldev***

shows the real (*rdev*) or logical (*ldev*) device that is logged onto this virtual machine.

### **ON [*netname.*]*luname***

identifies an SNA/CCS terminal (controlled by the VTAM service machine) that is logged onto this virtual machine. It is the name taken from the VTAM definition of the terminal as an SNA logical unit. The *netname.* is a network qualifier followed by a period delimiter, and is only present when the the EXTended option is specified and the VSM supports network qualified names.

### **VSM *userid***

identifies the user ID of the VTAM service machine managing the device.

### **ON SYSC**

identifies the system console as the real device that is logged onto this virtual machine.

### **DISCONNECTED**

indicates that the real device is disconnected.

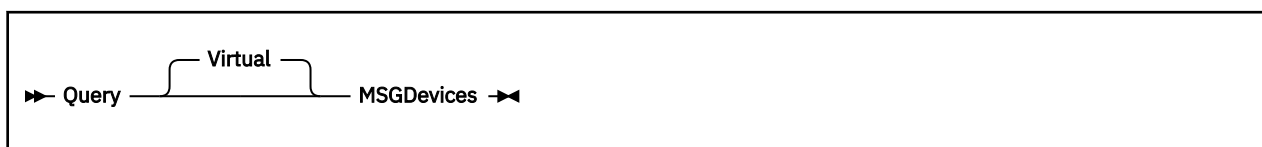
### ***vsub***

identifies the virtual subchannel number. This value is displayed only if the real device that is logged onto this virtual machine is other than the system console.

## **Messages**

- HCP003E Invalid option - *option*
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command.

## QUERY VIRTUAL MSGDEVICES



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL MSGDEVICES to display a list of all message devices within your I/O configuration.

### Usage Notes

1. To define a virtual message processor and associated virtual message devices, use the DEFINE MSGPROC command.
2. To display information about virtual message processors in your I/O configuration, use the QUERY VIRTUAL MSGPROC command.
3. To restart a virtual message processor that is in your I/O configuration, use the RESTART MSGPROC command.
4. To remove all message facility capability from your virtual machine, use the SET MSGFACIL OFF command.
5. To detach a virtual message processor and associated virtual message devices, use the DETACH MSGPROC command.

### Responses

Response 1:

To display a list of all message devices within your I/O configuration, enter the following:

```
query msgd
MSGD 0500 MESSAGE PROCESSOR = CFCC2   SUBCHANNEL = 0010
MSGD 0501 MESSAGE PROCESSOR = CFCC2   SUBCHANNEL = 0011
MSGD 0502 MESSAGE PROCESSOR = CFCC2   SUBCHANNEL = 0012
MSGD 0503 MESSAGE PROCESSOR = CFCC2   SUBCHANNEL = 0013
```

#### MSGD

indicates that the following four digit hex number is a message device number.

#### MESSAGE PROCESSOR

specifies the user ID of the message processor that is associated with the message device.

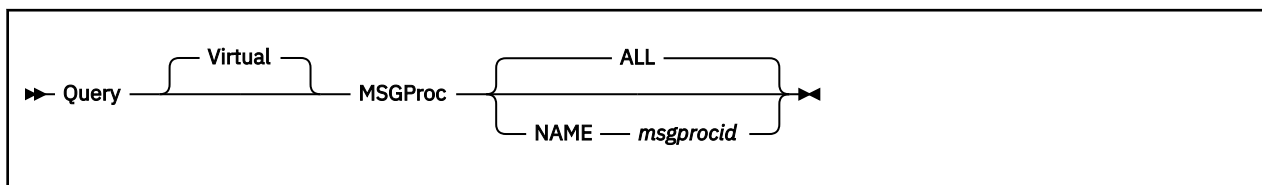
#### SUBCHANNEL

is the subchannel number for the message device.

### Messages

- HCP003E Invalid option – command contains extra option starting with *option*
- HCP040E Device MSGD does not exist

## QUERY VIRTUAL MSGPROC



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL MSGPROC to display information about virtual message processors within your I/O configuration.

### Operands

#### ALL

specifies that information about all CF Service Machines within your I/O configuration are to be displayed.

#### NAME *msgprocid*

specifies the user ID of a specific CF Service Machine within your I/O configuration to display.

### Usage Notes

1. To define a virtual message processor and associated virtual message devices, use the DEFINE MSGPROC command.
2. To display information about the virtual message devices within your I/O configuration, use the QUERY VIRTUAL MSGDEVICES command.
3. To restart a virtual message processor that is in your I/O configuration, use the RESTART MSGPROC command.
4. To remove all message facility capability from your virtual machine, use the SET MSGFACIL OFF command.
5. To detach a virtual message processor and associated virtual message devices, use the DETACH MSGPROC command.

### Responses

Response 1:

To display information for all message processors within your I/O configuration, enter the following:

```
query msgp
MESSAGE PROCESSOR=01      USERID=CFCC20K  USERS=01
TYPE=SIMDEV  MODEL=001  MFG=IBM  PLANT=EN  SEQ=00000CFCC20K
DEVICES=0103 0102 0101 0100
CGS=C0 80 (PA=C0) CFR=80 C0 (PA=C0)
CG=CFCCMNT

MESSAGE PROCESSOR=02      USERID=CFCC2      USERS=04
TYPE=SIMDEV  MODEL=001  MFG=IBM  PLANT=EN  SEQ=0000000CFCC2
DEVICES=0500 0501 0502 0503
CGS=C1 81 (PA=C0) CFR=81 C1
CG=SHARE1  SHARE2  SHARE4  SHARE3
Ready;
```

**MESSAGE PROCESSOR**

specifies the processor number assigned to the CF Service Machine. The first message processor defined by a user will be assigned as processor 1. Subsequent processors defined will be assigned the next lowest available number from two to eight.

**USERID**

is the user ID of the CF Service Machine that is running as a message processor.

**USERS**

is the total number of guests coupled to the message processor.

**TYPE**

is the six character device type in the worldwide unique ID of the message processors Node Descriptor

**MODEL**

is the three character model number in the worldwide unique ID of the message processors Node Descriptor.

**MFG**

is the three character code that identifies the manufacturer in the worldwide unique ID of the message processors Node Descriptor.

**PLANT**

is the two character code that identifies the plant of manufacture in the worldwide unique ID of the message processors Node Descriptor.

**SEQ**

is the twelve character code that contains the serial number in the worldwide unique ID of the message processors Node Descriptor. This field will make the Node Descriptor of the CF Service Machine unique in z/VM. The sequence field will contain the CF Service Machine user ID right justified. All leading or illegal characters will be padded with EBCDIC zeros.

**DEVICES**

is a list of message device numbers in the user's I/O configuration that represent the message processor.

**CGS**

(Coupled Guest Sender) is a list of message facility sender CHPIDS in the coupled guest's virtual I/O configuration, which are used by the devices specified in the DEVICES= list to communicate to the displayed message processor.

**CFR**

(Coupling Facility Receiver) is a list of message facility receiver CHPIDS in the message processor's virtual I/O configuration, which are used by the message processor to communicate to the devices specified in the DEVICES= list.

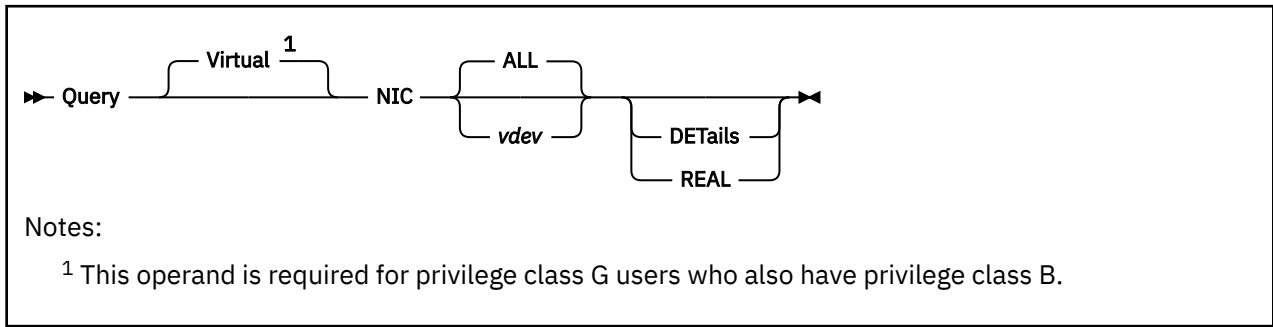
**CG**

is a list of all the coupled guests (CFUSER virtual machines) that are currently coupled to the message processor.

**Messages**

- HCP002E Invalid operand — *operand*
- HCP003E Invalid option — *option*
- HCP003E Invalid option — command contains extra option(s) starting with *option*
- HCP020E Userid missing or invalid
- HCP2805E Message processor *msgprocid* does not exist in your I/O configuration.

## QUERY VIRTUAL NIC



### Authorization

Privilege Class: G

### Purpose

Use QUERY NIC to display information about a network device in your virtual machine configuration. The response for each network adapter includes:

- Adapter configuration (base device address, number of devices)
- Adapter type for example, HIPER for HiperSockets
- Assigned MAC address and associated attributes.
- Status of each associated virtual device for example, DETAILS option

### Operands

#### ALL

indicates a request for "all" simulated network interface cards (NIC) in the invoker's virtual machine configuration

#### vdev

specifies the virtual device number (hex) of the adapter to be examined. You may specify the virtual device number of any virtual device that is attached to the adapter. However, CP responses will refer to each virtual adapter by its *base* device address; this is the first device in the range.

#### DETAils

indicates a request for full details about the selected adapters. The response includes the following information for each adapter:

- the role (CTL-READ, CTL-WRITE, or DATA) assigned to this device
- the IP Addresses registered to this device
- the priority for all packets sent from this virtual NIC's network connection to an external network if the virtual NIC is connected to a virtual switch. The priority value (PQUPLINKTX) will be used when priority queuing is enabled on the virtual switch (PRIQueueing ON).

#### REAL

indicates a request for a display of real network devices attached to the invoker's virtual machine.

### Examples

#### 1. QUERY NIC:

Display a list of simulated network interface cards belonging to this virtual machine:

```
cp query v nic
Adapter 0500.P00 Type: QDIO      Name: UNASSIGNED  Devices: 3
```

```

MAC: 02-00-0F-00-01-4B      LAN: FARRELLS TEST
Adapter 05D0.P00 Type: HIPERS  Name: UNASSIGNED  Devices: 8
MAC: 02-00-0F-00-01-4E      LAN: * Internal      MFS: 163

```

This response describes a QDIO adapter at 0500 which is connected to a guest LAN, and a HiperSockets adapter at 05D0 which is using its internal LAN (indicated by LAN "\* Internal"). "UNASSIGNED" means your software has not yet assigned a name to this interface.

## 2. QUERY NIC DETAILS:

- Display detailed information about a simulated network interface card:

```

cp q v nic 2000 det
Adapter 2000.P00 Type: QDIO      Name: TCPIP20      Devices: 3
MAC: 02-00-00-00-00-13 LAN: SYSTEM SUBNTA
PQUplinkTX: Normal
RX Packets: 33521      Discarded: 0      Errors: 0
TX Packets: 24955      Discarded: 0      Errors: 0
RX Bytes: 34225670      TX Bytes: 33713275
Connection Name: z/VM0000 State: Session Established
Device: 2000 Unit: 000 Role: CTL-READ
Device: 2001 Unit: 001 Role: CTL-WRITE
Device: 2002 Unit: 002 Role: DATA      Port: 0065
VLAN: 0002 Assigned by user
Router: Primary IPv4 VLAN 0002
Options: Broadcast Multicast IPv4 VLAN
Unicast IP Addresses:
  9.60.59.32      MAC: 02-00-00-00-00-13
Multicast IP Addresses:
  224.0.0.1      MAC: 01-00-5E-00-00-01
  224.0.0.9      MAC: 01-00-5E-00-00-09

```

This response describes a QDIO adapter at 2000 which is connected to a guest LAN, and provides detailed information about each active connection group.

- Display detailed information about a device managed by DIAGNOSE Code X'2A8':

```

cp q v nic 0AAA det
Adapter 0AAA.P00 Type: QDIO      Name: UNASSIGNED  Devices: 3
MAC: 02-34-56-00-01-00      VSWITCH: SYSTEM TEST
PQUplinkTX: Normal
RX Packets: 2      Discarded: 0      Errors: 0
TX Packets: 4      Discarded: 0      Errors: 0
RX Bytes: 64      TX Bytes: 128
Connection Name: *DIAG2A8 State: Session Established
Device: 0AAA Unit: 000 Role: DATA-DIAG Port: 65
VLAN: 0001
Options: Ethernet Broadcast
Unicast MAC Addresses:
  02-34-56-00-01-00 IP: 10.1.6.3
  02-34-56-00-01-00 IP: 50C0:C216::10:1:6:3
Multicast MAC Addresses:
  01-00-00-00-00-01
  01-00-00-00-00-02
  01-00-00-00-00-03
  01-00-00-00-00-04
  01-00-00-00-00-05
Unassigned Devices:
Device: 0AAB Unit: 001 Role: Unassigned
Device: 0AAC Unit: 002 Role: Unassigned

```

For more information on [DIAGNOSE Code X'2A8'](#), see *z/VM: CP Programming Services*.

## 3. QUERY NIC REAL

Display a list of real network devices attached to the virtual machine:

```

cp q v nic real
Adapter 0400 Type: QDIO      Name: UNASSIGNED  Devices: 1
MAC: 02-15-24-00-00-09      LAN: * External
Adapter 0401 Type: QDIO      Name: UNASSIGNED  Devices: 1
Adapter 0402 Type: QDIO      Name: UNASSIGNED  Devices: 1

```

## 4. QUERY NIC 7000 DET

If you enter QUERY NIC 7000 DET, you will see something like this:

## QUERY VIRTUAL NIC

```
Adapter 7000.P00 Type: QDIO      Name: NIC_7000    Devices: 3
MAC: 02-22-96-00-00-04        VSWITCH: SYSTEM ETH1
MAC: 02-22-96-00-00-05        Device: 7000 Current Protected
PQUplinkTX: Normal
Data Counters Cleared: 9/24/15 02:35:40
Discarded Counters Cleared: 9/24/15 02:35:40
Error Counters Cleared: 9/24/15 02:35:40
RX Packets: 25                Discarded: 0      Errors: 0
TX Packets: 9                 Discarded: 2      Errors: 0
RX Bytes: 1086                TX Bytes: 378
Connection Name: HALLOLE      State: Session Established
Device: 7000 Unit: 000 Role: CTL-READ
Device: 7001 Unit: 001 Role: CTL-WRITE
Device: 7002 Unit: 002 Role: DATA Port: 0066
VLAN: 0001
Options: Ethernet Broadcast
Unicast MAC Addresses:
02-22-96-00-00-04
Multicast MAC Addresses:
01-00-5E-00-00-01 IP: 224.0.0.1
33-33-00-00-00-01 IP: FF02::1
33-33-FF-00-00-04 IP: FF02::FF00:4
```

## Responses

The **QUERY NIC** response includes optional lines. The general format is:

```
Adapter vdev.Pnn Type: type      Name: portname   Devices: devs
MAC: macaddr LAN: ownerid lanname MFS: mfsize
MAC: macaddr Device: dev macstatus
PQUplinkTX: priority
Data Counters Cleared: mm/dd/yy hh:mm:ss
Discarded Counters Cleared: mm/dd/yy hh:mm:ss
Error Counters Cleared: mm/dd/yy hh:mm:ss
RX packets: rpackets            Discarded: nnn      Errors: nnn
TX packets: tpackets            Discarded: nnn      Errors: nnn
RX bytes: rbytes                TX bytes: tbytes
Connection Name: groupid       State: description
Device: vdev Unit: nnn Role: role Port: portnum
VLAN: nnnn status
Router: type ipversion vlan
Options: options
Unicast IP Addresses:
ipaddr MAC: macaddr <flags>
ipaddr MAC: macaddr <flags>
Multicast IP Addresses:
mpaddr MAC: macaddr <flags>
mpaddr MAC: macaddr <flags>
Device: (repeated for each device in the group)
...
Connection Name: ...
Unassigned Devices:
Device: vdev Unit: nnn Role: Unassigned
Device: (repeated for each unassigned device)
```

### Adapter

*vdev.Pnn* is the base virtual device address and port number of the simulated NIC.

### Type

*type* is the adapter type (QDIO or HIPERS).

### Name

*portname* is the adapter (or port) name assigned by your communications software. The value "UNASSIGNED" is displayed if your software does not assign a value.

### Devices

*devs* is the number of devices that form this adapter.

### LAN

*ownerid lanname* uniquely identifies the coupled guest LAN (or displays "\*" internal" for an uncoupled adapter).

### MFS

*mfssize* indicates the Maximum Frame Size. It is displayed only for HiperSockets adapters.



This line may contain a value of "SYSTEM" for the MAC address. That indicates that a SYSTEM MAC was requested, but has not yet been allocated.

**Current**

indicates a MAC address that is currently assigned to the device address.

**Pending**

indicates a MAC address that is pending. When the device is re-initialized, the pending MAC address will become current.

**PQUplinkTX**

indicates the priority for all packets sent from a virtual NIC's network connection to an external network. The PQUplinkTX value will be used when the virtual NIC is connected to a virtual switch, and priority queuing is enabled on the virtual switch (PRIQueuing ON).

**LOW**

specifies that outbound traffic to the external network will be sent at a low priority. This traffic will use the low priority queue which is serviced less frequently than the normal or high priority queues.

**NORMAL**

specifies that outbound traffic to the external network will be sent at a normal priority. This traffic will use the normal priority queue which is serviced less frequently than the high priority queue but more frequently than the low priority queue.

**HIGH**

specifies that outbound traffic to the external network will be sent at a high priority. This traffic will use the high priority queue which is serviced more frequently than the normal or low priority queues.

**RX packets**

*rxpackets* is the number of packets received.

**TX packets**

*txpackets* is the number of packets transmitted.

**Discarded**

*nnn* indicates the number of packets discarded.

**Errors**

*nnn* indicates the number of error packets discarded.

**RX bytes**

*rxbytes* is the number of bytes received.

**TX bytes**

*txbytes* is the number of bytes transmitted.

**Connection Name**

*groupid* is the name assigned by your communications software to this connection.

**State**

*description* represents the current state of this connection (usually "Session Established").

**MAC: mm-mm-mm-mm-mm-mm Device: vdev macstatus port**

Device MAC address and attributes line

This line will be displayed for each network device defined on the NIC adapter that has a MAC address assigned by the system administrator. If no MAC address assignments are made, then a network connection established on any device associated with this NIC will use the adapter MAC address. The adapter MAC address is the first MAC address displayed following the "Adapter" line (line 2).

**MAC: mm-mm-mm-mm-mm-mm**

Mac address assigned.

**Device vdev**

is the virtual device number assigned to the MAC address.

### ***macstatus***

is the current status of the specified MAC address.

#### **Current**

indicates the MAC address is currently assigned.

#### **Pending**

indicates the MAC address is waiting to be assigned the next time a network connection is reestablished.

### **Data Counters Cleared**

Indicates that the transmitted and received counters were reset to zero via the SET VSWITCH COUNTERS CLEAR DATA command at the time stated in the message.

### **Discarded Counters Cleared**

Indicates that the discarded counters were reset to zero via the SET VSWITCH COUNTERS CLEAR DISCARDED command at the time stated in the message.

### **Error Counters Cleared**

Indicates that the errors counters were reset to zero via the SET VSWITCH COUNTERS CLEAR ERRORS command at the time stated in the message.

### ***port***

is the current status of the specified MAC address.

#### **Protected**

when displayed indicates the MAC address is protected and cannot be overridden by the guest when establishing a network connection. If not displayed, then it is an unrestricted MAC address and may be overridden by a guest.

### **Unit**

*nnn* indicates the relative device number on this adapter (the base device is unit 000).

### **Role**

*role* is the role assigned to this device (CTL-READ, CTL-WRITE, or DATA).

### **Port**

*portnum* is the port number associated with the guest connection or the virtual switch network connection.

### **VLAN**

*nnnn(n)* indicates the effective VLAN ID or IDs for the device. Multiple VLAN ID lines may be shown if the virtual switch is protected by an ESM.

*status* is "Assigned by system" or "Assigned by user". If the specified VLAN ID conflicts with the authorizations for this user, "Conflicting VLAN ID" is displayed.

### **Router**

*type ipversion* indicates the adapter has configured itself as a router with type Primary, Secondary or Multicast in IP version IPv4 or IPv6. *vlan* is either VLAN followed by the VLAN ID, if it is a VLAN-specific router, or Global, if it is a global router. If it is a global router, lines will follow to describe which VLAN IDs the adapter is routing for.

The router line is displayed only when a router is configured.

### **Options**

*options* indicates the currently enabled features for this interface, such as Broadcast, Multicast, IPv4 or IPv6.

### ***ipaddr***

represents one Unicast IP address that has been registered to this device and optional information relating to the IP address.

An IPv4 address is displayed in dotted-decimal format, for example 9 . 130 . 44 . 144.

An IPv6 address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros). For example, the IPv6 address

```
F802:0000:0000:0000:0000:0000:0A00:0001
```

is displayed in compressed format as `F802::A00:1`.

### ***mpaddr***

represents one Multicast IP Address.

For IPv4, the address is displayed in dotted-decimal format with the associated MAC address.

For IPv6, the address is displayed in compressed format ("::" designates multiple 16-bit groups of zeros).

### **MAC**

*macaddr* represents a MAC address.

### ***flags***

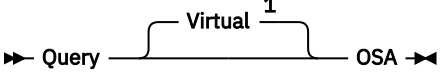
additional information that may be displayed with the IP address. These include:

- If the simulated NIC is coupled to a virtual switch, local indicates that this address was registered in a virtual switch LAN, but is NOT currently registered in the external network. This may indicate the external interface is disconnected. It may also indicate that this address is only supported within the immediate network (as it is in the case of an IPv6 address within an IP Layer VSWITCH).
- Conflict indicates that this address was registered locally but cannot be used at all because of a conflict with the network external to the virtual switch LAN.
- Remote indicates that this address was registered automatically to represent a source address found in a datagram. This entry is automatically removed after the IP Timeout (IPTIMEOUT) has expired.
- Manual indicates that this address was registered manually by an ARP cache control command from the driver. This entry will be automatically removed after the IP Timeout (IPTIMEOUT) has expired, or it may be removed earlier by the driver.

### **Messages**

- HCP006E Invalid device type *vdev*
- HCP040E Device {*vdev*|NIC} does not exist

## QUERY VIRTUAL OSA



Notes:

<sup>1</sup> This operand is required for privilege class G users who also have privilege class B.

### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL OSA to display status of the Open Systems Adapter (OSA) devices accessible to your virtual machine.

### Responses

For each OSA device dedicated to your virtual machine configuration, the following is displayed:

```
OSA vdev {ON DEV rdev } [SUBCHANNEL = vsub] [(OSA AGENT)] [TOKEN = token]
      {ON NIC vdev AS UNIT unit}
      vdev DEVTYP devtyp CHPID chp chptyp
```

For each OSA device that uses the Queued Direct I/O (QDIO) hardware, the following is displayed:

```
vdev {QDIO-ELIGIBLE } [(QEBSM)]
     {QDIO PENDING ACTIVE}
     {QDIO ACTIVE }
     {QIOASSIST status }
```

If the subchannel for the OSA device is in either QDIO PENDING ACTIVE or QDIO ACTIVE state, the following line is displayed for each input and output data queue that was established by a program running in a V=V virtual machine only:

```
vdev type sync priority IOCNT = nnnnnnnn ADP = nnn PROG = nnn UNAVAIL = nnn
vdev BYTES = nnnnnnnnnnnnnnnnn
```

*nnn*

If a virtual MAC address is assigned, the following is displayed:

```
vdev MAC macaddr status
```

The terms shown in the above Results have the following meanings:

**vdev**

is the virtual device number of the OSA device.

**rdev**

is the real device number of the OSA device.

**SUBCHANNEL = vsub**

identifies the virtual subchannel number.

**OSA AGENT**

indicates that the device is an OSA Agent.

**DEVTYPE *devtyp***

identifies the device type of the OSA device.

**CHPID *chp chptyp***

identifies the CHPID number and CHPID type of the OSA device.

**TOKEN = *token***

identifies the subchannel token assigned to the QDIO connection when it is managed by QDIO enhanced buffer state management (QEBSM).

**QDIO-ELIGIBLE**

indicates that the device is eligible to use the QDIO Hardware Facility for direct data transfer.

**QDIO PENDING ACTIVE**

indicates that a program running in the virtual machine has established QDIO data queues for transferring data between the program and the specified OSA device. Although the data queues are created, they must be activated by the program running in the virtual machine before data transfer can commence.

**QDIO ACTIVE**

indicates that a program running in the virtual machine has activated the QDIO data queues. Direct data transfer between the specified OSA device and the program may now occur.

**QIOASSIST *status***

The status is one of the following:

- **QIOASSIST NOT AVAILABLE** – indicates that the processor does not support the queue-I/O assist (QDIO performance assist for V=V guests), or z/VM is running in an environment where the queue-I/O assist is not available. See the SET QIOASSIST command for a list of requirements to use queue-I/O assist.
- **QIOASSIST- ELIGIBLE** – indicates the device is eligible to use the queue-I/O assist (QDIO performance assist for V=V guests) when the virtual machine begins using the device with no traces in effect.
- **QIOASSIST DISABLED** – indicates the device is not permitted to use the queue-I/O assist (QDIO performance assist for V=V guests). The device must have been attached to the virtual machine with the NOQIOAssist option either on the CP ATTACH command or the system-directory DEDICATE statement.
- **QIOASSIST ACTIVE** – indicates the queue-I/O assist (QDIO performance assist for V=V guests) is active and in-use.

**QEBSM**

indicates that the device connection is managed using enhanced QDIO buffer state management protocol. When QEBSM manages a QDIO connection, a QDIO I/O request can be initiated without z/VM involvement.

***type***

indicates the type of QDIO data queue:

**INP**

Input queue, used to transfer data from the OSA device to the program

**OUT**

Output queue, used to transfer data from the program to the OSA device

***sync***

indicates whether the program and CP versions of the QDIO data queue are currently synchronized:

+

Synchronized

-

Not synchronized

For V=V virtual machines, CP translates the guest's virtual environment to the real hardware. On behalf of the guest, CP makes a replica, or shadow, of the QDIO data queues created by a program

## QUERY VIRTUAL OSA

running in the virtual machine. CP gives the shadow queues to the hardware to use. CP is then responsible for keeping the data queues synchronized between the program and the hardware.

### **priority**

indicates the priority of the QDIO data queue. All input and output queues are listed according to their queue priority, with the lowest number queues having the highest priority.

### **IOCNT = nnnnnnnn**

specifies the number of data transfers that have completed since the QDIO data queue was last activated by the program, in decimal. CP increments the IOCNT each time a QDIO data buffer changes from an adapter-owned state to a program-owned state.

### **BYTES = nnnnnnnnnnnnnnnn**

specifies the number of bytes transferred on that queue since device activation, in hexadecimal.

### **ADP = nnn**

specifies the number of QDIO data buffers currently pending or available for a direct data transfer. This is the number of QDIO data buffers currently owned by the adapter.

### **PROG = nnn**

specifies the number of QDIO data buffers currently owned by the program running in the virtual machine that have storage blocks allocated.

### **UNAVAIL = nnn**

specifies the number of QDIOw data buffers currently owned by the program running in the virtual machine that have no storage blocks allocated.

### **MAC macaddr**

specifies the MAC address assigned.

*Status* contains one of the following:

#### **Current**

indicates a MAC address that is currently assigned to the device address.

#### **Pending**

indicates a MAC address that is pending. When the device is re-initialized, the pending MAC address will become current.

This example displays three virtual OSA devices that are part of a single virtual network adapter:

## Examples

Example 1:

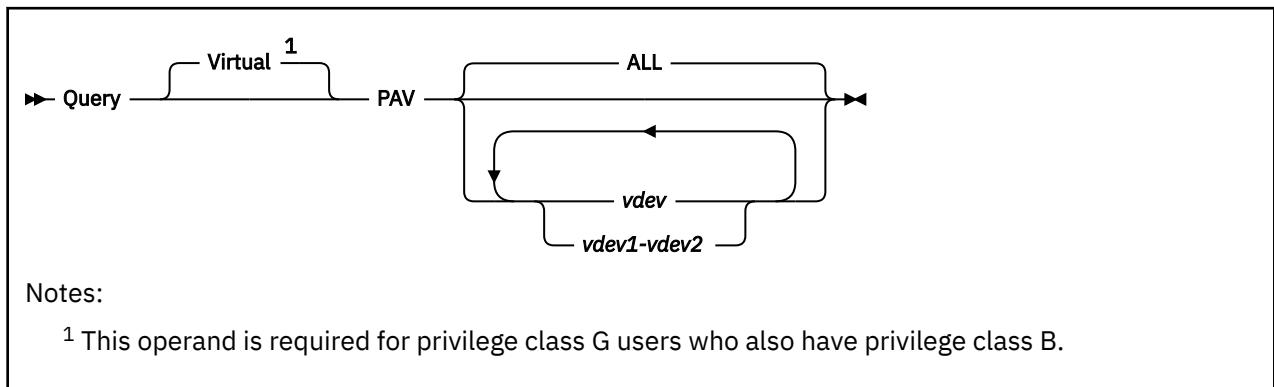
If you enter QUERY VIRTUAL OSA, you might see something like this:

```
cp query virtual osa
OSA 0500 ON NIC 0500 UNIT 000 SUBCHANNEL = 000D
      0500 DEVTYPE OSA          CHPID 80 OSD
0500 MAC 02-00-00-00-00-05 CURRENT
      0500 QDIO-ELIGIBLE
OSA 0501 ON NIC 0500 UNIT 001 SUBCHANNEL = 000E
      0501 DEVTYPE OSA          CHPID 80 OSD
      0501 QDIO-ELIGIBLE
OSA 0502 ON NIC 0500 UNIT 002 SUBCHANNEL = 000F
      0502 DEVTYPE OSA          CHPID 80 OSD
      0502 QDIO-ELIGIBLE
```

## Messages

- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP263E Too many operands were supplied for this command

## QUERY VIRTUAL PAV



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL PAV to display information about the Parallel Access Volume (PAV) devices and the HyperParallel Access Volume (HyperPAV) devices accessible to your virtual machine.

### Operands

#### ALL

tells CP to display a list of all PAV and HyperPAV devices accessible to your virtual machine.

#### *vdev*

#### *vdev1-vdev2*

is the virtual device address of the device for which you want PAV or HyperPAV information. You can specify a single device address, a range of device addresses, or any combination.

### Usage Notes

1. Use the QUERY VIRTUAL *vdev* DETAILS command or the QUERY VIRTUAL DASD DETAILS command to display all the virtual alias Parallel Access Volumes associated with a particular virtual base Parallel Access Volume.

### Responses

Response 1:

If you enter QUERY V PAV 300, and virtual device 300 is a dedicated PAV base device, you see a response like this:

```
q v pav 300
PAV BASE 0300 ON E100 RWS001
Ready;
```

This response tells you that virtual PAV base 300 is defined on dedicated real PAV base E100 with label RWS001.

Response 2:

If you enter QUERY V PAV 4300, and virtual device 4300 is a dedicated HyperPAV base device, you see a response like this:

## QUERY VIRTUAL PAV

```
q v pav 4300
HYPERPAV BASE 4300 on F900 YAC001 POOL 2
Ready;
```

This response tells you that virtual HyperPAV base 4300 is defined on dedicated real HyperPAV base F900 with label YAC001 which resides in HyperPAV pool number 2.

Response 3:

If you enter QUERY V PAV 301, and virtual device 301 is a dedicated PAV alias device, you see a response like this:

```
q v pav 301
PAV ALIAS 0301 on E149 RWS001 FOR BASE 0300
Ready;
```

This response tells you that virtual PAV alias 301 is defined on dedicated real PAV alias E149 with label RWS001. It also tells you that alias 301 is associated with PAV base 300.

Response 4:

If you enter QUERY V PAV 4301, and virtual device 4301 is a dedicated HyperPAV alias device, you see a response like this:

```
q v pav 4301
HYPERPAV ALIAS 4301 ON F981 POOL 2
Ready;
```

This response tells you that virtual HyperPAV alias 4301 is defined on dedicated real HyperPAV alias F981 which resides in HyperPAV pool number 2.

Response 5:

If you enter QUERY V PAV 680, and virtual device 680 is a PAV base device defined as a minidisk or temporary disk, you see a response like this:

```
q v pav 680
PAV BASE 0680 ON E101 RWS002 ASSIGNED E101
Ready;
```

This response tells you that virtual PAV base 680 is defined on, and assigned to, real base E101 with label RWS002.

Response 6:

If you enter QUERY V PAV 4680, and virtual device 4680 is a HyperPAV base device defined as a minidisk or temporary disk, you see a response like this:

```
q v pav 4680
HYPERPAV BASE 4680 ON F901 YAC002 ASSIGNED F901 POOL 4
Ready;
```

This response tells you that virtual HyperPAV base 4680 is defined on and assigned to real HyperPAV base F901 with label YAC002 which resides in HyperPAV pool number 4.

Response 7:

If you enter QUERY V PAV 681, and virtual device 681 is a PAV alias device defined as a minidisk or temporary disk, you see a response like this:

```
q v pav 681
PAV ALIAS 0681 ON E101 RWS002 ASSIGNED E14F FOR BASE 0680
Ready;
```

This response tells you that virtual PAV alias 681 is defined on real PAV base E101 with label RWS002. It also tells you that it is assigned to real PAV alias E14F and is associated with virtual PAV base 680.

Response 8:



If you enter QUERY V PAV 4681, and virtual device 4681 is a HyperPAV alias device defined as a minidisk or temporary disk, you see a response like this:

```
q v pav 4681
HYPERPAV ALIAS 4681 ASSIGNED F991 POOL 4
Ready;
```

This response tells you that virtual HyperPAV alias 4681 is assigned to real HyperPAV alias F991 which resides in HyperPAV pool number 4.

Response 9:

If you enter a QUERY V PAV command specifying specific devices and/or device ranges and none of those devices are PAV or HyperPAV devices, you see a response like this:

```
q v pav 999 A00-A10
NO VIRTUAL PAV DEVICES
Ready;
```

Response 10:

If you enter QUERY V PAV ALL, you see a response like this:

```
q v pav all
PAV BASE 0300 ON E100 RWS001
PAV ALIAS 0301 ON E149 RWS001 FOR BASE 0300
PAV ALIAS 0302 ON E14A RWS001 FOR BASE 0300
PAV ALIAS 0303 ON E14B RWS001 FOR BASE 0300
PAV ALIAS 0304 ON E14C RWS001 FOR BASE 0300
PAV BASE 0680 ON E101 RWS002 ASSIGNED E101
PAV ALIAS 0681 ON E101 RWS002 ASSIGNED E14F FOR BASE 0680
PAV ALIAS 0682 ON E101 RWS002 ASSIGNED E150 FOR BASE 0680
PAV ALIAS 0683 ON E101 RWS002 ASSIGNED E14D FOR BASE 0680
PAV ALIAS 0684 ON E101 RWS002 ASSIGNED E14E FOR BASE 0680
HYPERPAV BASE 4300 ON F900 YAC001 POOL 2
HYPERPAV ALIAS 4301 ON F981 POOL 2
HYPERPAV ALIAS 4302 ON F982 POOL 2
HYPERPAV ALIAS 4303 ON F983 POOL 2
HYPERPAV ALIAS 4304 ON F984 POOL 2
HYPERPAV BASE 4680 ON F901 YAC002 POOL 4
HYPERPAV ALIAS 4681 ASSIGNED F991 POOL 4
HYPERPAV ALIAS 4682 ASSIGNED F992 POOL 4
HYPERPAV ALIAS 4683 ASSIGNED F993 POOL 4
HYPERPAV ALIAS 4684 ASSIGNED F994 POOL 4
Ready;
```

Response 11:

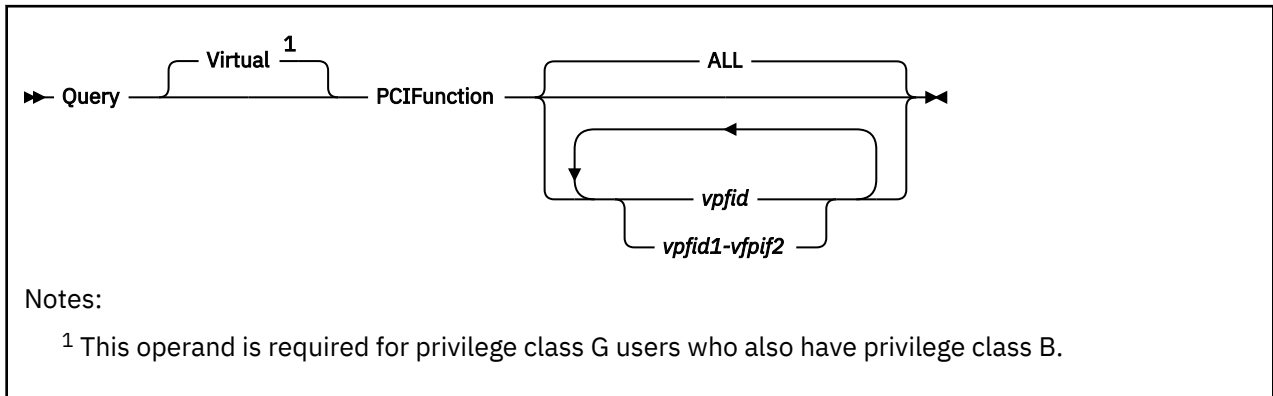
If you enter a QUERY V PAV ALL command and there are no PAV or HyperPAV devices in your configuration, you see a response like this:

```
q v pav all
NO VIRTUAL PAV DEVICES
Ready;
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

## QUERY VIRTUAL PCIFUNCTION



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL PCIFUNCTION to display the status of PCI functions accessible to your virtual machine.

### Operands

#### ALL

specifies all virtual PCI functions accessible to your virtual machine

#### *vpfid*

#### *vpfid1-vpfid2*

specifies a virtual PCI function ID or a range of virtual PCI function IDs whose status is to be queried. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. The total number of virtual PCI functions specified cannot exceed 256.

### Responses

Response 1:

If you enter QUERY VIRTUAL PCIFUNCTION, you will see something like:

```
PCIF 00000003 ON 00000021 ENABLED 10GbE RoCE
PCIF 00000004 ON 00000046 DISABLED 10GbE RoCE
PCIF 00000005 ON 00000058 ENABLED 10GbE RoCE
PCIF 00000006 ON 00000024 ERROR 10GbE RoCE
```

The first number is the virtual PCI function ID. The next number is the real PCI function ID followed by the state of the PCI function and the type of PCI function.

The state of the function is one of the following:

#### **BLOCKED**

PCI function blocked

#### **DISABLED**

PCI function available for use by the configuration but not enabled

#### **ENABLED**

PCI function available for use by the configuration and enabled

**ERROR**

Error detected by z/PCI facility

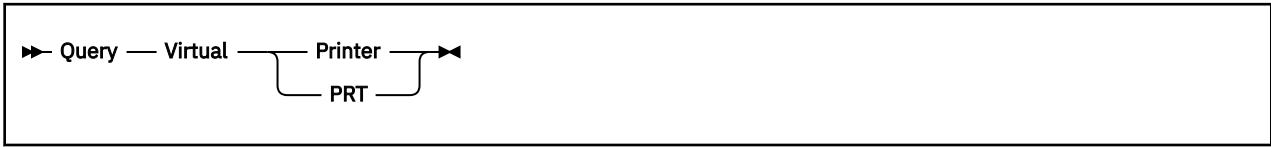
**PERMERR**

Error recovery for PCI function unsuccessful

**Messages**

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP022E A virtual PCI function ID was not supplied or it was invalid.
- HCP040E PCI function *vpfid* does not exist
- HCP1003E *vpfid* not found.
- HCP5780E Adjunct does not support the *command subcommand* command
- HCP6000E The range of PCI function ids cannot exceed 256.

## QUERY VIRTUAL PRINTER



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL PRINTER to display the status of all print devices accessible to your virtual machine.

### Usage Notes

1. VAFP spool files are intended for use by AFP printers and real printers managed by RSCS. VAFP printers are not intended for normal use as a virtual printer.

### Responses

Response 1:

For **each virtual printer** defined in your virtual machine configuration, the first three lines of the response are:

```
PRT  vdev CL c { CONT} { HOLD} {COPY nnn} { READY} FORM userform
          {NOCONT} {NOHOLD} {CPY *nnn} {NOTREADY}

vdev TO userid {RDR} DIST distcode FLASHC ccc DEST dest
          {PRT}
          {PUN}

vdev FLASH name CHAR char [char[char[char]]] MDFY cmod n FCB fcb LPP nnn
```

For **3800 printers only**, an additional line follows:

```
vdev SIZE width length {2WCGM} {BTS} {DATCK }
          {4WCGM} {CFS} {NODATCK}
```

The remaining lines of the response for **all printers** are:

```
vdev type { EOF} {OPEN spoolid} { KEEP} { MSG} { NAME fname ftype}
          {NOEOF} {CLOSED } {NOKEEP} {NOMSG} {NONAME }

[vdev SUBCHANNEL = vsub]
```

#### **vdev**

identifies the virtual printer.

#### **CL c**

identifies the spool file class given to spool files created on this device.

#### **CONT**

#### **NOCONT**

controls the continuous spooling option. CONT causes all CLOSE requests to be ignored unless specified with the EOF override option.

**HOLD****NOHOLD**

controls the HOLD status. HOLD prevents files from being printed on a real device by putting a user hold status on the file.

**COPY *nnn*****CPY *\*nnn***

identifies the number of copies requested for the spool file. This is shown as a 3-digit number (*nnn*). The asterisk (\*) indicates that the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page. If the asterisk (\*) is not present, the entire file is printed sequentially.

**READY****NOTREADY**

indicates whether the device is in ready or not-ready status. The status can be changed by the READY and NOTREADY commands.

**FORM *userform***

displays the form associated with the device and assigned to the spool files as they are created.

**TO *userid***

identifies the virtual machine that receives output from your virtual printer.

**RDR****PRT****PUN**

identifies the spool file queue to receive your printer output.

**DIST *distcode***

identifies the distribution code assigned to each spool file produced from the device.

**FLASHC *ccc***

identifies the number of copies printed while the forms overlay frame is superimposed on the output pages. The default is *ccc*=000.

**DEST *dest***

identifies the destination value to be assigned to each spool file created on the device. If the destination value has not been changed by a SPOOL command, the default is OFF.

**FLASH *name***

identifies the name of the forms overlay frame superimposed on the output pages if the file is printed on the 3800 printer.

**CHAR *char***

identifies the name of the character set or sets used to generate output if the file is printed on the 3800 printer.

**MDFY *cmod***

identifies the copy modification module used to alter output text if the file is printed on the 3800 printer.

***n***

identifies the copy modification character set number. If no copy modification character is set, *n* is 0.

**FCB *fc***

identifies the forms control buffer used in the vertical formatting of a page if the file is printed on the 3800 printer.

**LPP *nnn***

identifies the lines per page setting for the virtual printer. The value can be a decimal number in the range 30 to 255, or 'OFF', which indicates that internal defaults will be used.

***width***

indicates the form width code (in hexadecimal) for a virtual 3800 printer and is a 2-character field.

***length***

identifies, in decimal notation, the length of the paper in half-inches loaded into a 3800 printer.

**2WCGM**

**4WCGM**

specifies the number of writable character generation modules (WCGM) for the virtual 3800 printer. A WCGM is a 64-position portion of the 3800's character generation storage that holds the scan elements of one character set.

**BTS**

**CFS**

designates the stacker assumed for the virtual 3800 printer. BTS is a burster-trimmer-stacker, and CFS is a continuous forms stacker.

**DATCK**

**NODATCK**

specifies the handling of certain virtual 3800 data checks. If DATCK is specified, all 3800 data checks are reflected to the virtual machine (provided the BLOCK DATA CHECK CCW has not been issued). If NODATCK is specified, only data checks that occur because of invalid translate table specifications or unmatched FCB codes are reflected to the virtual machine.

**type**

identifies an IBM printer type or VAFP printer.

**EOF**

**NOEOF**

EOF closes, or NOEOF leaves open, the spool file on the specified device regardless of the CONT setting for the device.

**OPEN spoolid**

**CLOSED**

indicates that a spool file has been created on the console and that the file is still open (OPEN), or indicates that no spool file is open on the device (CLOSED).

**KEEP**

**NOKEEP**

controls the keep status, which indicates whether the current spool file is to be placed in user hold after processing.

The keep status is placed on the spool file. After it is transcribed at the real printer, the file is assigned user hold status and is not purged. The file's keep status is retained.

**MSG**

**NOMSG**

indicates whether you want to be informed about file printing and punching. If you specify MSG, you are informed. The message is sent to you when the file begins either printing or punching on a real device. This message is not sent if you have issued SET IMSG OFF. Specify NOMSG to reset the MSG option.

**NAME *fname ftype***

**NONAME**

identifies the file name and file type, if any, to be assigned to spool files created on the device.

***vsub***

identifies the virtual subchannel number.

Response 2:

```
PRT vdev ON PRT rdev [SUBCHANNEL = vsub]
```

is returned when the printer is a real device dedicated to your virtual machine.

***vdev***

identifies the virtual device number.

***rdev***

identifies the real device number.

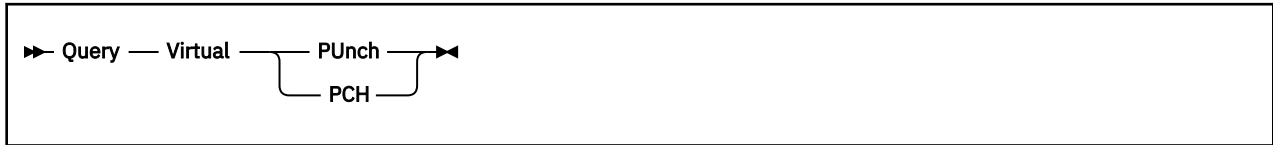
***vsub***

identifies the virtual subchannel number.

**Messages**

- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command

## QUERY VIRTUAL PUNCH



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL PUNCH to display the status of all punch devices accessible to your virtual machine.

### Responses

Response 1:

```
PUN  vdev CL a { CONT} { HOLD} {COPY nnn} { READY} FORM userform
      {NOCONT} {NOHOLD} {CPY *nnn} {NOTREADY}

vdev TO userid {RDR} DIST distcode DEST dest
              {PRT}
              {PUN}

vdev FLASH name ccc CHAR char [char[char[char]]] MDY cmod n FCB fcb

vdev type { EOF} {OPEN spid} { KEEP} { MSG} { NAME fname ftype}
          {NOEOF} {CLOSED } {NOKEEP} {NOMSG} {NONAME }

[vdev SUBCHANNEL = vsub]
```

is returned for each of your virtual punches.

Response 2:

```
PUN vdev ON PUN rdev [SUBCHANNEL = vsub]
```

is returned when the punch is a real device dedicated to your virtual machine.

#### **vdev**

identifies the virtual device number.

#### **CL a**

identifies the spool file class assigned to spool files created on this punch.

#### **CONT**

#### **NOCONT**

controls the continuous spooling option. CONT causes all CLOSE requests to be ignored unless specified with the EOF override option.

#### **HOLD**

#### **NOHOLD**

controls the hold status. HOLD prevents files from being punched on a real device by putting a user hold status on the file. If you specify NOHOLD, you override the spool device HOLD option and prevent the file from being placed in hold status.

#### **COPY nnn**

#### **CPY \*nnn**

identifies the number of copies requested for the spool file. This is shown as a 3-digit number (*nnn*). An asterisk (\*) indicates that the 3800 printer prints each page from the spooled output files *nnn*.



times before going on to the next page. If the asterisk (\*) is not present, the entire file is printed sequentially.

**READY****NOTREADY**

indicates whether the device is in a ready or not-ready status. You can change the status with the READY and NOTREADY commands.

**FORM *userform***

displays the form associated with the device and assigned to the spool files as they are created.

**TO *userid***

identifies the virtual machine that receives output from your virtual punch.

**RDR****PRT****PUN**

identifies the spool file queue to receive your punch output.

**DIST *distcode***

identifies the spool file distribution code.

**DEST *dest***

identifies the destination value to be assigned to each spool file created on the device. If the destination value has not been changed by a SPOOL command, the default is OFF.

**FLASH *name ccc***

*name* is the name of the forms overlay frame superimposed on the output pages if the file is printed on a 3800 printer. *ccc* is the number of copies printed while the forms overlay frame is superimposed on the output pages. The default for *ccc* is 000.

**CHAR *char ...***

identifies the character sets used to generate output if the file is printed on a 3800 printer.

**MDFY *cmod***

identifies the copy modification module used to alter output text if the file is printed on a 3800 printer.

***n***

identifies the copy modification character set number. If no copy modification character is set, *n* is 0.

**FCB *fc***

identifies the forms control buffer used in the vertical formatting of a page if the file is printed on a 3800 printer.

***type***

identifies an IBM punch type.

**EOF****NOEOF**

closes (EOF), or leaves open (NOEOF), the spool file on the specified device regardless of the CONT setting for the device.

**OPEN *spid*****CLOSED**

indicates that a spool file (*spid*) has been created on the console and that the file is still open (OPEN), or indicates that no spool file is open on the device (CLOSED).

**KEEP****NOKEEP**

controls the keep status, which indicates whether the current spool file is to be placed in user hold after processing.

The keep status is placed on the spool file. After it is transcribed at the real punch, the file is assigned user hold status and is not purged. The file's keep status is retained. Specify NOKEEP to override the current KEEP setting for the device so that KEEP is not set on the spool file.

**MSG****NOMSG**

indicates whether you want to be informed about file printing and punching. If you specify MSG, you are informed of file printing and punching. The message is sent to you when the file begins either printing or punching on a real device. This message is not sent if you have issued SET IMSG OFF. Specify NOMSG to reset the MSG option.

***rdev***

identifies the real device number.

**NAME *fname ftype*****NONAME**

indicates whether a file name and file type are to be assigned to the spool file. If a name is to be assigned, it is also displayed.

***vsub***

identifies the virtual subchannel number.

### Messages

- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command

## QUERY VIRTUAL READER



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL READER to display the status of all readers accessible to your virtual machine.

### Responses

Response 1:

```

RDR  vdev CL a   { CONT } { HOLD } { EOF } { READY }
           { NOCONT } { NOHOLD } { NOEOF } { NOTREADY }

           vdev type      { OPEN spid }
                           { CLOSED }

{ KEEP } { RESCAN } [SUBCHANNEL = vsub]
{ NOKEEP } { NORESCAN }
  
```

is returned for each of your virtual readers.

Response 2:

```
RDR vdev ON RDR rdev [SUBCHANNEL = vsub]
```

is returned when the reader is a real device dedicated to your virtual machine.

#### **vdev**

is the virtual device number.

#### **CL a**

is the spool file class that your reader services; your reader can only read files of a matching class. A class of asterisk (\*) indicates that your reader can read files of any class.

#### **type**

is an IBM reader device type.

#### **CONT**

#### **NOCONT**

controls the continuous spooling option. CONT specifies that reading continues without end-of-file indications at the end of each virtual spool file until the virtual device is empty. NOCONT specifies that the end-of-file is signaled after each spool file is read, and then the reader is reset at the next logical file.

#### **HOLD**

#### **NOHOLD**

controls the HOLD status. The HOLD/NOHOLD options override the setting for the virtual device or the spool file. The file is not put in HOLD status unless the KEEP option is also specified. If you specify HOLD, the file is not purged after processing and remains eligible for the device. If you specify NOHOLD, the file is purged unless the KEEP option is also specified.

**EOF****NOEOF**

EOF closes, and NOEOF does not close, the spool file on the specified device, regardless of the CONT setting for the device.

**READY****NOTREADY**

indicates whether the device is in a ready or not-ready status. To change the status, use the READY and NOTREADY commands.

**OPEN spid****CLOSED**

indicates the status of the spool file. If the status is open, the system spool file ID is also displayed.

**KEEP****NOKEEP**

controls the keep status, which indicates whether the current spool file is to be placed in user hold after processing. KEEP overrides the keep setting on either the virtual reader or the spool file. After processing, the file is placed in hold status, and the KEEP option on the file remains unchanged. Specify NOKEEP to override either file or device keep characteristics. The disposition of a file closed with NOKEEP (whether it is purged or retained) is controlled by the hold status of the virtual reader. If it were closed KEEP, it would never be purged.

**RESCAN****NORESCAN**

When a reader is spooled CONT, RESCAN causes the first logical file in the virtual reader queue to always be the next file read. This allows files which have been ordered or changed during continuous processing to become eligible for processing. NORESCAN causes scanning to resume at the next logical file in the virtual reader queue.

A reader cannot be spooled CONT, RESCAN, HOLD, and NOKEEP, as this would cause the same file to be processed repeatedly. If a SPOOL command is issued which would result in the reader being spooled with this combination, the reader will be spooled KEEP to remove the conflict.

***vsub***

is the virtual subchannel number.

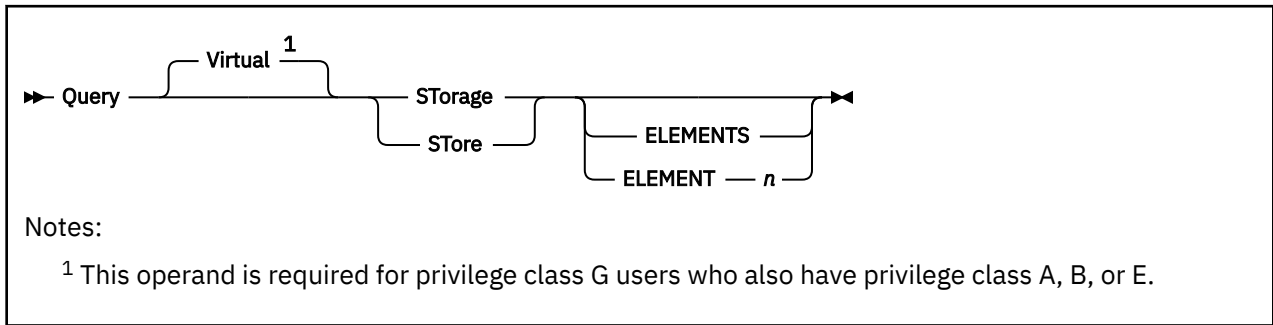
***rdev***

is the real device number of the reader.

**Messages**

- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command

## QUERY VIRTUAL STORAGE/STORE



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL STORAGE or QUERY VIRTUAL STORE to display information about the storage accessible to your virtual machine.

If the virtual storage configuration was defined with reserved or standby storage, then dynamic storage reconfiguration (DSR) is supported. Otherwise, DSR is not supported. If DSR is supported, then virtual storage is comprised of storage element 0 and storage element 1.

### Operands

#### ELEMENTS

displays information about storage elements 0 and 1.

#### ELEMENT *n*

displays information about a specific storage element, where *n* is either 0 or 1.

### Responses

Response 1:

This response is issued if no reserved or standby storage has been defined since the last system reset:

```
STORAGE = nu
```

where:

***n***

is a 1- to 7-digit decimal number.

***u***

is the storage unit suffix:

**M**

megabytes

**G**

gigabytes

**T**

terabytes

**P**

petabytes

**E**

exabytes

Response 2:

This response is issued if reserved or standby storage has been defined since the last system reset. The first line is a summary. If you specified the ELEMENT *n* or ELEMENTS operand, one or two additional lines are displayed to provide details about the storage elements.

```
STORAGE = nu MAX = nu INC = nu STANDBY = nu RESERVED = nu
[ELEMENT 0 ATTACHED ASSIGNED = nu STANDBY = nu RESERVED = nu]
[ELEMENT 1 {ATTACHED} ASSIGNED = nu STANDBY = nu RESERVED = nu]
           {STANDBY }
           {RESERVED}
```

where:

**STORAGE = nu**

specifies the size of online (assigned) virtual machine storage.

**MAX = nu**

specifies the maximum size of virtual machine storage as defined in the user directory entry for the virtual machine.

**INC = nu**

specifies the size of the virtual machine storage increment.

**STANDBY = nu**

specifies the amount of virtual machine storage in standby state that is available to be configured online.

**RESERVED = nu**

specifies the amount of virtual machine storage in reserved state that can be defined online or as standby.

**ELEMENT 0 ATTACHED**

**ELEMENT 1 ATTACHED**

**ELEMENT 1 STANDBY**

**ELEMENT 1 RESERVED**

specifies the state of the storage element:

**ATTACHED**

The specified storage is available to the guest. Storage element 0 is always attached.

**STANDBY**

Storage element 1 is in detached-standby state and not available to the guest. The guest can issue an instruction to attach the element.

**RESERVED**

Storage element 1 is in detached-reserved state and not available to the guest. To make the storage available, the virtual machine console operator must issue the DEFINE STORAGE STANDBY command to increase the size of the virtual machine's standby storage and change the state of storage element 1 to detached-standby. The guest can then issue an instruction to attach the element.

**ASSIGNED = nu**

specifies the amount of storage in the storage element that is online (assigned to the virtual machine).

Response 3:

This response is issued if CONFIGURATION was specified on DEFINE STORAGE:

```
STORAGE = nu
Storage Configuration:
configspec

Extent Specification          Address Range
-----
0.sssssssssssssu 00000000_00000000 - endadres_endadres
```

```
boundaryaddressu.ssssssssssssu begadres_begadres - endadres_endadres
boundaryadru-nnnnu.ssssssssssssu begadres_begadres - endadres_endadres
```

where:

**STORAGE = nu**

specifies the size of online (assigned) virtual machine storage, including the configuration extents.

**configspeg**

is the storage configuration specification input used on the DEFINE STORAGE CONFIGURATION command that defines the current storage configuration.

**Extent Specification**

specifies the information supplied by the DEFINE STORAGE command. Each configuration extent is listed on a separate line:

**boundaryaddressu**

**boundaryadru**

is the boundary address as specified in the DEFINE STORAGE command, where *u* is the 1-character storage unit suffix.

**sssssssssssssu**

is the extent size, 1 - 14 decimal digits followed by the 1-character storage unit suffix.

**nnnnu**

is the decrement specified in DEFINE STORAGE used to modify the extent start boundary address, expressed as 1 - 4 decimal digits followed by the 1-character storage unit suffix (which must be a smaller storage unit than that used for the start boundary it modifies). The decrement is subtracted from the base start boundary specification to determine the effective extent start boundary address.

**Address Range**

specifies the extent information converted to beginning and ending addresses:

**begadres\_begadres**

is the 16-digit extent start address expressed in hexadecimal (always ends in X'00000', a megabyte boundary). Underscores are used for readability if SET UNDERSCORE ON has been issued.

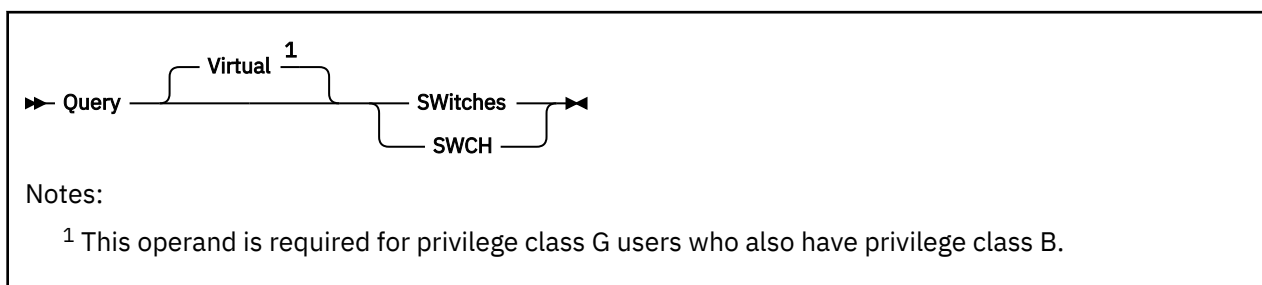
**endadres\_endadres**

is the 16-digit extent end address expressed in hexadecimal (always ends in X'FFFFFF', one less than a megabyte boundary). Underscores are used for readability if SET UNDERSCORE ON has been issued.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid

## QUERY VIRTUAL SWITCHES



### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL SWITCHES to display the status of all your virtual switch devices.

### Operands

#### SWitches

#### SWCH

displays the status of your virtual switch devices.

### Responses

Response 1:

```
SWCH vdev ON SWCH rdev {SUBCHANNEL = vsub}
```

is displayed for each switch device attached to your virtual machine system

#### **vdev**

is the virtual device number.

#### **rdev**

is the real device number

#### **vsub**

is the virtual subchannel number.

### Messages

- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP263E Too many operands were supplied for this command.



## QUERY VIRTUAL SYSASCII

►► Query — Virtual 1 — SYSAscii ►►

### Notes:

<sup>1</sup> This operand is required for privilege class G users who also have privilege class B.

### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL SYSASCII to display the status of the ASCII console in your virtual machine configuration.

### Responses

SYSASCII *status*

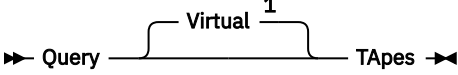
#### ***status***

is either *active* or *inactive*, corresponding to whether the ASCII console session on the Hardware Management Console (HMC) is open or not open.

### Messages

- HCP121E SYSASCII not attached to you
- HCP263E Too many operands were supplied for this command.

## QUERY VIRTUAL TAPES



Notes:

<sup>1</sup> This operand is required for privilege class G users who also have privilege class B.

### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL TAPES to display the status of all tape devices accessible to your virtual machine.

### Responses

Response 1:

```
TAPE vdev ON DEV rdev type [GIVEN BY userid] {R/O }
[GIVEN TO userid] {R/W*}

[SUBCHANNEL = vsub] [NOASSIGN]

[SPXTAPE operation status]

[ENCRYPTION CAPABLE]
```

is displayed for each tape device attached to your virtual machine.

Note that (\*) this is the mode that the giver currently has the tape drive in and is *not* the mode in which it was given.

#### **vdev**

identifies the virtual device number.

#### **rdev**

identifies the real device number.

#### **type**

identifies the device type of the attached tape device.

#### **GIVEN BY *userid***

#### **GIVEN TO *userid***

indicates that the real device was given to another user ID or was received from another user ID through the GIVE command. These parameters only appear for tape drives that have been given with the return option specified.

#### **R/O**

#### **R/W**

identifies the type of access you have to the device. R/O indicates that you have read only access; R/W indicates that you have both read and write access.

#### **SUBCHANNEL = *vsub***

identifies the virtual subchannel number.

#### **NOASSIGN**

indicates the tape drive was attached with the NOASSIGN option.

**SPXTAPE**

indicates an SPXTAPE operation is running on the virtual tape drive.

**operation**

is the SPXTAPE operation: DUMP, LOAD, or SCAN.

**status**

is the status of the SPXTAPE operation:

**ACTIVE**

Read (LOAD or SCAN) or write (DUMP) operations are currently being done to the tape drive.

**MOUNT**

The tape drive is not active because SPXTAPE is waiting for a tape to be mounted.

**APPEND**

The SPXTAPE command that started the operation was entered with the APPEND option, and the append sequence is not complete. Enter another SPXTAPE command (addressing the same tape devices) with the APPEND option to select more files, or enter the command without the APPEND option to complete the append. If the operation is LOAD or SCAN, actual loading or scanning of files cannot begin until the append sequence is complete.

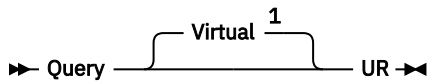
**ENCRYPTION CAPABLE**

indicates the tape device supports hardware encryption.

**Messages**

- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command.

## QUERY VIRTUAL UR



Notes:

- <sup>1</sup> This operand is required for privilege class G who also have privilege class B.

### Authorization

Privilege Class: G

### Purpose

Use QUERY VIRTUAL UR to display the status of all unit record devices (printers, punches, and readers) accessible to your virtual machine.

### Responses

See responses under [“QUERY VIRTUAL PRINTER”](#) on page 1324, [“QUERY VIRTUAL PUNCH”](#) on page 1328, and [“QUERY VIRTUAL READER”](#) on page 1331.

### Messages

- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command

## QUERY VMDUMP



### Authorization

Privilege Class: G

### Purpose

Use QUERY VMDUMP to identify the output mode of VMDUMP and any open VMDUMP spool file.

### Operands

#### MAP

indicates that a map of the address ranges of guest storage in the open VMDUMP file should be displayed, in addition to the other command responses.

### Usage Notes

1. If a continuous mode dump file was open when noncontinuous mode is set, that file remains open until completion of the next VMDUMP or DIAGNOSE X'94' request that writes to the file. For this reason, it is possible to receive the response VMDUMP NOCONT as well as responses describing an open, continuous-mode VMDUMP file.
2. The file type of the open VMDUMP file may not match that displayed from QUERY READER. This may occur if the FORMAT option was used after the spool file was initially open. The file type is not changed until the spool file is completed and closed. The file type shown in response to QUERY VMDUMP is the type given the dump file when it is closed.
3. Use SET UNDERSCORE to control whether an underscore character is used to separate 16-digit addresses for the VMDUMP command.

### Examples

```
q vmd map
VMDUMP CONT
File 0018 VMDUMP RSCSV2 is open for output
*no output on SNA printer
The VMDUMP address ranges are:
00000000-0000BFFF 0001A000-000FFFFF 00200000-003CDFFF 00EF0000-00EF4FFF
00F00000-01088FFF 02000000-02FFFFFF

For ASIT 1B543C7563DF6923
SPACEID JOE:SPACE1
Format: RSCSV2
Dump identifier: *no output on SNA printer
The VMDUMP address ranges are:
00001000-00002FFF 00004000-00004FFF 0001C000-0001FFFF 00021000-0003AFFF
00051000-00060FFF 000BB000-000CDFFF

For ASIT 1A23B5678C912D45
SPACEID JOE:SPACE2
Format: FILE
Dump identifier: none
The VMDUMP address ranges are:
00001000-00001FFF 0000B000-0000FFFF
```

## Responses

Response 1:

```
VMDUMP NOCONT
```

is returned when the virtual machine is not in continuous output mode for VMDUMP.

Response 2:

```
VMDUMP CONT
```

is returned when the virtual machine is in continuous output mode for VMDUMP.

Response 3:

The following additional lines are returned for a QUERY VMDUMP or QUERY VMDUMP MAP command when an open output file exists (from one or more prior VMDUMP requests while in continuous output mode). The following second line is returned only if the dump has a *dumpid*:

```
File spid VMDUMP filetype is open for output
*dumpid
```

### ***spid***

identifies the spool file.

### ***filetype***

identifies the virtual machine type (*filetype*) supplied by the FORMAT operand of the VMDUMP command.

### **\**dumpid***

identifies the dump. It is preceded by an asterisk (\*).

For each address space in the dump, these lines are displayed:

```
For ASIT asit
  SPACEID owner:space_name
Format:      format
Dump identifier: *dumpid
```

### ***asit***

is the ASIT of the address space.

### ***owner:***

is the user ID that owns the address space.

### ***space\_name***

is the name of the address space. The space name is a string of alphanumeric and national characters, 1 to 24 characters long.

A header line and additional response lines are returned for a QUERY VMDUMP MAP command when an open output file exists.

For virtual machines in ESA/390 mode, each line maps from one to four address ranges:

```
The VMDUMP address ranges are:
xxxxxxxx-yyyyyyyy xxxxxxxx-yyyyyyyy . . .
:
```

### ***xxxxxxxx-yyyyyyyy***

identifies a hexadecimal address range of guest real storage contained in the VMDUMP spool file.

For virtual machines in z/Architecture mode, each line maps from one to two address ranges:

```
The VMDUMP address ranges are:
xxxxxxxx_xxxxxxxx-yyyyyyyy_yyyyyyyy xxxxxxxx_xxxxxxxx-yyyyyyyy_yyyyyyyy
:
```

**xxxxxxxx\_xxxxxxxxx-yyyyyyyy-yyyyyyyy**

identifies a hexadecimal address range of guest real storage contained in the VMDUMP spool file.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP260E {Command|*command* command} processing cannot complete due to errors.
- HCP410E CP ENTERED; PAGING ERROR

## QUERY VMLAN

► Query — VMLAN ◄

### Authorization

Privilege Class: G

### Purpose

Use QUERY VMLAN to determine the current status of guest LAN activity on the CP system. The response indicates the current value of the system-wide guest LAN attributes, and the number of guest LANs and virtual switches currently defined in the system.

Use the QUERY LAN command to obtain information about specific guest LANs. Use the QUERY VSWITCH command to obtain information about specific virtual switches.

### Usage Notes

System-wide attributes for guest LAN or virtual switch support can be set using the [VMLAN Statement](#) in the SYSTEM CONFIG file. For more information, see in *z/VM: CP Planning and Administration*. These attributes can be changed during normal system operation by using the SET VMLAN command. See [“SET VMLAN” on page 1752](#).

### Examples

Display a typical guest LAN response:

```
query vmlan
  VMLAN maintenance level:
    Latest Service: VM65918
  VMLAN MAC address assignment:
    System MAC Protection: OFF
    MACADDR Prefix: 023456 USER Prefix: 023456
    MACIDRANGE SYSTEM: 800000-801000
                      USER: 800000-800100

  VMLAN default accounting status:
    SYSTEM Accounting: OFF   USER Accounting:OFF
  VMLAN general activity:
    PERSISTENT Limit: 20     Current: 15
    TRANSIENT Limit: 10     Current: 0
  Trace Pages: 8

  VMLAN Directory Network Authorization: ENABLED
  IVL Domain: A   MAC address: 03-FF-FF-FF-FF-01   VLAN: NONE
  IVL Domain Heartbeat Timeout: 30
  IVL Domain Capability: 8000000000000000
  Member: CASEY   MAC address: 02-34-56-00-00-05
    State: Active
    Port Group Status: Synchronized
    Global VSWITCH Status: Synchronized
    Heartbeat Count: <local>
    Member Capability: C000000000000000 Maintenance level: V642

  Member: JONES   MAC address: 02-34-77-00-00-08
    State: Active
    Port Group Status: Synchronized
    Global VSWITCH Status: Synchronized
    Heartbeat Count: 20
    Member Capability: 8000000000000000 Maintenance level: V632
```



## Responses

The normal response is:

```
VMLAN maintenance level:
Latest Service: update
VMLAN MAC address assignment
System MAC Protection: setting
MACADDR Prefix: prefix  USER Prefix: prefix
MACIDRANGE SYSTEM: range
                USER: range

VMLAN default accounting status:
SYSTEM Accounting: onoff  USER Accounting: onoff
VMLAN general activity
PERSISTENT Limit: max           Current: count
TRANSIENT Limit: max           Current: count
Trace Pages: nn

VMLAN Directory Network Authorization: DISABLED | ENABLED
IVL Domain: x  MAC address: macid  VLAN: [ NONE | vlanid]
IVL Domain Heartbeat Timeout: timeout
IVL Domain Capability: capabilitystring
Member: membername  MAC address: macid
State: state
Port Group Status: pgstatus
Global VSWITCH Status: gvstatus
Heartbeat Count: hbcount
Member Capability: capabilitystring  Maintenance level: mntid
```

### **update**

is the most recent update that affects guest LAN and virtual switch simulation.

### **System MAC Protection**

*setting* indicates the level of MAC address protection level to be used for virtual and real networks whose MAC address protection settings are not specified (set to UNSPECIFIED).

### **prefix**

is the prefix used for MAC addresses that are automatically assigned by CP to a network device. This value (MACPREFIX) is set by a VMLAN statement in the SYSTEM CONFIG file.

### **USER prefix**

is the prefix used for MAC addresses explicitly assigned by a user from CP's system directory. This value is set by a VMLAN (USERPREFIX) statement in the SYSTEM CONFIG file.

### **range(s)**

are the ranges defined to be used for the MACID portion of MAC address generation. These values (MACIDRANGE SYSTEM and MACIDRANGE USER) are set by a VMLAN statement in the SYSTEM CONFIG file.

The MACIDRANGE SYSTEM and USER range specified in the VMLAN statement are valid only when MACADDR and USER prefix values are identical. When a different USER prefix value is specified, the entire MACIDRANGE is used for both automatic and user defined locally administered MAC addresses on the system. As a result the MACIDRANGE specified on the VMLAN statement is ignored. The SYSTEM range is set to 000001-FFFFFF and the USER range is set to 000000-000000.

### **onoff**

indicates whether this feature is **ON** (enabled) or **OFF** (disabled).

### **max**

represents the preset limit for this category.

### **count**

represents the actual number of guest LANs and virtual switches in this system.

### **Trace Pages**

indicates the number of pages allocated for an internal trace table to keep track of general networking events.

### **VMLAN Directory Network Authorization: *DISABLED | ENABLED***

indicates whether Directory Network Authorization is enabled (default setting) or disabled. See [“SET VMLAN”](#) on page 1752.

**IVL Domain: x**

identifies the domain to which the virtual switch belongs, where x is a letter, A-H. Global networking objects such as a global virtual switch can span system running z/VMs that are members of the same IVL domain.

When no IVL Virtual Switch has been defined, the system is not a member of an IVL domain. The IVL domain line displayed is the following, and no additional lines are shown.

```
IVL Domain: None
```

When the IVL Virtual Switch has been defined, but there is a system configuration which prevents the z/VM image from becoming an active IVL member, the following line is displayed. For more information on the reason, see [z/VM: CP Messages and Codes](#).

```
IVL Domain: Isolated          Reason: HCPIVLxxxxE
```

**macid**

identifies a MAC address used to communicate with IVL domain members. The IVL domain MAC address is a multicast address used to communicate with all members. The Member MAC address is a unicast MAC address to communicate with a specific member.

**VLAN [NONE | *vlanid*]**

identifies the VLAN, if any, to which the IVL domain belongs.

**IVL Domain Heartbeat Timeout: *timeout***

identifies the length of time that is allowed to expire before the local system declares a communication problem with another IVL domain member if heartbeat signals are missing. Every 5 seconds, each IVL member broad casts a signal to confirm communications with all IVL domain members.

**IVL Domain Capability: *capabilitystring***

identifies the minimum capability of all members.

**Member *membername***

identifies the 8 byte EBCDIC z/VM system identifier of the domain member.

**State *state***

is the local system's view of the other member's state.

**Inactive**

The state indicates the member was previously known to the local system, but has not responded to the most recent communication attempts.

**Pending Leave**

This transitional state indicates the member is leaving the domain.

**Error**

An error was encountered when the member attempted to join the domain.

**Pending Join**

This transitional state indicates the member is joining the domain.

**Active**

The member is active in the IVL domain.

**Port Group Status *pgstatus***

describes whether or not the overall shared port group definitions are compatible. The Port Group Status line is displayed only when a shared port group has been defined.

**Synchronized**

The local system has exchanged messages with the domain member and determined that the domain member's configuration of all shared port groups matches the shared port groups defined on the local system.

**Pending Synchronization**

The local system is checking to make sure that configuration of the shared port group matches on the remote system.

**Not Synchronized**

The local system's shared port group configuration does not match the other members configuration for one or more shared port groups. Use the QUERY PORT GROUP command with the DETAILS option to display details about configuration mismatches.

**Global VSWITCH Status *gvstatus***

describes whether or not the global virtual switch definitions are compatible. The Global VSWITCH Status line is displayed only when a global virtual switch has been defined.

**Synchronized**

The local system has exchanged messages with the domain member and determined that the domain member's configuration of all global virtual switches matches the global virtual switches defined on the local system.

**Pending Synchronization**

The local system is checking to make sure that configuration of the global virtual switch matches on the remote system.

**Not Synchronized**

The local system's global virtual switch configuration does not match the other members configuration for one or more global virtual switches. Use the QUERY VSWITCH command with the DETAILS option to display details about configuration mismatches.

**Heartbeat Count *hbcount***

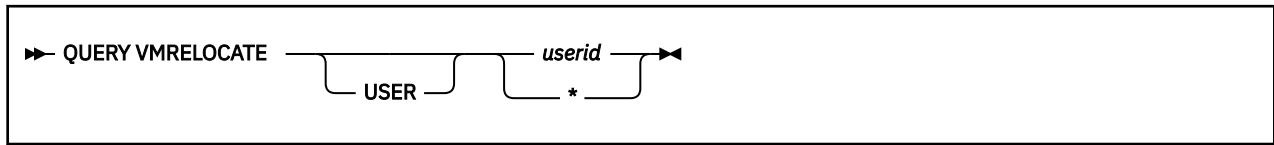
represents the number of times a heartbeat message has been received from the domain member.

**Member Capability: *capabilitystring* Maintenance Level: *mntid***

identifies the capability of individual members, and maintenance level id of the system. As z/VM enhancements are made IVL domain members may be using different maintenance levels or releases of z/VM. The *capabilitystring* for each system indicates which enhancements are available on that IVL domain member.

```
8000000000000000 - Base capability
C000000000000000 - Collaborative Load Balance Support
```

## QUERY VMRELOCATE



### Authorization

Privilege Class: A, B, C, E

### Purpose

Use the QUERY VMRELOCATE command to display the relocation capability of the specified user. This command can be used to determine to which relocation domain a particular user belongs, and therefore, to which SSI members it may be relocated.

### Operands

#### USER *userid*

specifies the user ID whose relocation capability is being queried. The user ID must be logged on. If "\*" is specified, the target user is the command issuer.

### Usage Notes

1. Use the QUERY RELODOMAIN command to determine which member systems are included in the user's relocation domain.
2. If the QUERY VMRELOCATE command is issued for a user ID defined by a multiconfiguration virtual machine definition, which cannot be relocated, the response will indicate the single-member domain of the system where the command is issued, with relocation disabled. The relocation capability of such a user cannot be changed with the SET VMRELOCATE command.
3. For information about [Using Relocation Domains](#), see *z/VM: CP Planning and Administration*.

### Responses

Response 1:

```
Running on member sysname
Relocation enabled|disabled in Domain domain
```

Response 2:

```
This system is not a member of an SSI cluster
```

### Messages

- HCP0002E Invalid operand - *operand*
- HCP0003E Invalid option - *option*
- HCP0013E Conflicting option - *option*
- HCP0026E Operand missing or invalid
- HCP0045E *userid* not logged on
- HCP1818I Excluded members: *system\_names*
- HCP1831E The command could not be completed because of a temporary condition. Try again later.
- HCP6706E Invalid string - *string*

## QUERY VMSG/PVMSG



### Authorization

Privilege Class: G

### Purpose

Use QUERY VMSG/PVMSG to recall the last 16 messages that your virtual machine operating system sent to the system console.

The z/VM control program, as part of its service processor simulation, intercepts these messages and displays them on the virtual machine operator console.

For operating system messages requiring an operator response, use the VINPUT command to direct the z/VM control program to send data to the operating system running in your virtual machine as though the data had been entered from the system console.

### Operands

#### VMSG

requests a list of the last 16 nonpriority messages that the operating system running in your virtual machine sent to the system console.

#### PVMSG

requests a list of the last 16 priority messages which the operating system running in your virtual machine sent to the system console.

### Responses

Response 1:

```
1 [H] [Prompt text:] message text1
2 [H] [Prompt text:] message text2
...
nnn [H] [Prompt text:] message text nnn
```

where messages 1 through *nnn* are the last 16 messages (of the type requested) that the operating system running in your virtual machine sent to the system console. They are displayed in the order of their receipt. Note that the list does not include messages deleted by the virtual machine console operator or by the operating system running in your virtual machine.

The H is displayed on an operating system message designated by the operating system to be “held” until it has been deleted by the virtual machine console operator or by the operating system running in your virtual machine. These messages are displayed even though they may not be within the last 16 messages intercepted.

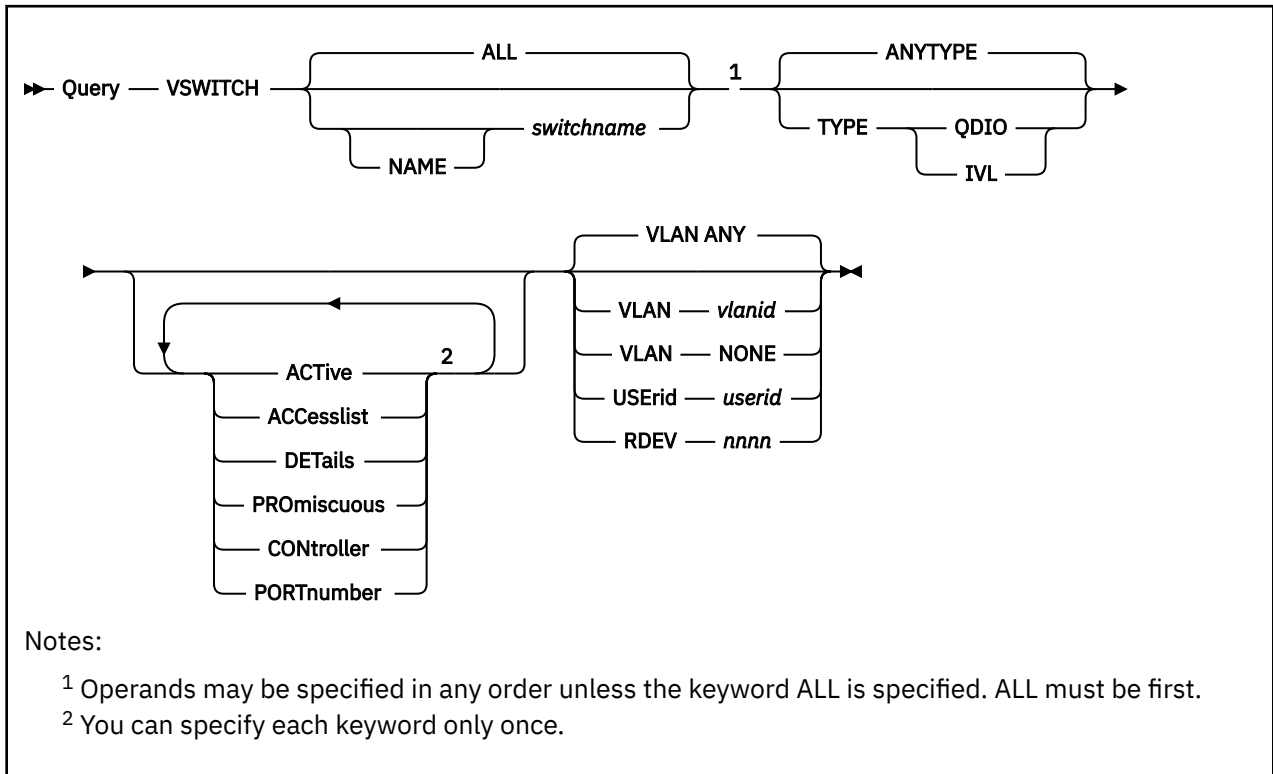
“Prompt text” identifies message text the operating system has designated as a prompt, associated with a related message.

**Note:** The number preceding each message may be used with VDELETE to identify the messages to be deleted from the list of messages sent to the system console.

### Messages

- HCP002E Invalid operand - *operand*
- HCP6530I There are no {priority|non-priority} messages currently outstanding.

# QUERY VSWITCH



## Authorization

Privilege Class: B, G

## Purpose

Use **QUERY VSWITCH** to display information about the designated virtual switch, or all virtual switches.

A class G user can view information about a virtual switch if the user is in the CP access list for the virtual switch, or if the user currently has an adapter connected to it.

A class B user can view information about any virtual switch in the system.

## Operands

### ALL

displays information for all possible virtual switches. This is the default value when an explicit *NAME switchname* is omitted.

### ANYTYPE

indicates a request for any virtual switch regardless of LAN TYPE. This is the default when the TYPE operand is omitted.

### TYPE QDIO

### TYPE IVL

restricts the responses to virtual switches of the specified type.

### NAME *switchname*

restricts the response to the virtual switch with the specified *switchname*. Use of the NAME keyword is required to restrict the response if the virtual switch name matches any of the QUERY VSWITCH keywords.

## QUERY VSWITCH

### ACTive

displays information about each active connection and each HiperSockets Bridge Capable Port. Each active connection represents a simulated network interface card (NIC) that has been connected to the virtual switch. The information displayed includes:

- Identity of the virtual switch.
- The list of users connected to each virtual switch, including the VLAN IDs each is authorized to use.
- IP addresses registered to the adapter.

Each HiperSockets Bridge Capable Port represents a guest on the HiperSocket channel that is bridge capable and is being bridged by the virtual switch.

### ACCesslist|PORTnumber

displays a list of ports that are authorized to connect to each virtual switch with a simulated network adapter.

This is the list pre-authorized through the use of the SET VSWITCH command with GRANT or PORTNUMBER operands, or currently connected as authorized by the NICDEF directory statement or an External Security Manager (ESM).

**Note:** When a NICDEF directory statement is used to authorize access to the VSWITCH, the list only includes the active connections. The user directory must be examined to get a complete access list. When the virtual switch is protected by an External Security Manager (ESM) the ESM interfaces must be used to determine the true authorization list.

### DETAils

displays all available information for each virtual switch. This is similar to specifying ACTIVE, except the display includes users that are coupled, but not active.

For global virtual switches, the display includes information about how each member of the global virtual switch is configured on each IVL domain member.

### PRomiscuous

displays a list of ports that are authorized for promiscuous mode on the virtual switch.

This is the list pre-authorized through the use of the SET VSWITCH command with GRANT or PORTNUMBER operands, or currently connected as authorized by the NICDEF directory statement or an External Security Manager (ESM).

### CONtroller

displays the list of user IDs that can control the OSA-Express devices associated with the virtual switch.

### USERid *userid*

displays the information as requested by other specified options, along with the subset of QUERY VSWITCH DETAILS that applies to either the named guest or HiperSockets Bridge Capable Port. An error is displayed if the guest *userid* is not coupled.

### RDEV *nnnn*

displays the information as requested by other specified options. Only the subset of QUERY VSWITCH DETAILS that applies to the matching Uplink or Bridge device is displayed. An error is displayed if device *nnnn* does not exist.

### VLAN *vlanid*

displays information restricted to connections (guest connections and HiperSockets Bridge Capable Ports) that have the specified *vlanid* active. **VLAN ANY** displays all connections regardless of any VLAN settings. **VLAN NONE** displays connections that have not enabled VLAN support.

## Usage Notes

**QUERY VSWITCH** can display the following state values for either an UPLINK or Bridge Port connection:

### Defined

The virtual switch is created, but no real device address has been specified.



**Defined-NOUPLINK**

The virtual switch is created, but the NOUPLINK option has been specified.

**Controller not available**

A z/VM TCP/IP virtual switch controller is not available to manage the device. The connection to the network is not operational.

**Disconnected**

The virtual switch is created, but the connection to the network is not operational. An operator stopped operation of the device using the TCP/IP STOP command, or there was an error during device initialization, such as a portname conflict or primary router setting. This case is also shown when the connection to the OSA-Express is lost.

**Disconnected – operator**

The virtual switch is created, but the connection to the network is not operational and requires operator intervention. The virtual switch was defined with the DISCONNECT option, or an operator stopped operation of the device using the SET VSWITCH DISCONNECT command.

**Devices attached**

The devices are attached to a virtual switch controller, but the connection to the network is not operational.

**Initialization in progress**

Initialization of the devices has begun. The connection to the real hardware LAN is not operational.

**Device not ready**

Initialization has completed, but the virtual switch controller received a response from the device indicating that the device is not ready. The connection to the network is not operational.

The virtual switch will wait a short period of time for the device to become ready. If that is not successful, the virtual switch will attempt to establish the network connection on a different device if one exists that reported a recoverable error condition.

**Ready**

The connection to the network is operational.

**Device disconnect in progress**

The connection to the network is not operational.

**Detach VSWITCH pending**

The virtual switch is being removed from the system.

**In Error Recovery**

Because of a failure, the virtual switch has started the process to move to a backup device and controller. Data transfer has been suspended for the connection, but will resume when recovery is complete.

**Restarting**

Because a failure occurred and no operational backup device was available, the virtual switch is attempting automatic restart of the failing device. Data transfer has been suspended for the connection, but will resume when the auto restart completes successfully.

The virtual switch will wait a short period of time for the device to recover. If that is not successful, the virtual switch will attempt to establish the network connection on a different device if one exists that reported a recoverable error condition.

**NOUPLINK**

There is no UPLINK port on this virtual switch.

**Examples****1. QUERY VSWITCH:**

```
cp query vswitch uvsw
VSWITCH SYSTEM UVSW Type: QDIO Connected: 0 Maxconn: INFINITE
PERSISTENT RESTRICTED NONROUTER Accounting: OFF
USERBASED LOCAL
VLAN Unaware
```

## QUERY VSWITCH

```
MAC address: 02-6F-5C-60-00-0B   MAC protection: UNSPECIFIED
IPTimeout: 5                     QueueStorage: 8
Isolation Status: OFF           VEPA Status: OFF
Unicast IP address count: 0
Uplink Port:
State: Ready                    PriQueueing: OFF
PMTUD setting: EXTERNAL        PMTUD value: 8992   Trace Pages: 8
RDEV: 3B03.P00 VDEV: 3B03 Controller: CONTROL1 ACTIVE
EQID: PRODNET1
RDEV: 1E26.P00 VDEV: 1E26 Controller: CONTROL3 BACKUP
EQID: PRODNET1
```

## 2. QUERY VSWITCH DETAILS:

This example displays all the available detail for VSWITCH VSWITCH3.

```
cp q vswitch vswitch3 det
VSWITCH SYSTEM VSWITCH3 Type: QDIO   Connected: 3   Maxconn: INFINITE
PERSISTENT RESTRICTED NONROUTER     Accounting: ON
USERBASED LOCAL
VLAN Aware Default VLAN: 0001   Default Porttype: Trunk   GVRP: Enabled
Native VLAN: 0001   VLAN Counters: OFF
MAC address: 02-22-33-00-00-0B   MAC Protection: Unspecified
IPTimeout: 5                     QueueStorage: 8
Isolation Status: OFF           VEPA Status: OFF
Unicast IP address count: 10
Uplink Port:
State: Ready                    PriQueueing: ON
PMTUD setting: EXTERNAL        PMTUD value: 8992   Trace Pages: 8
RDEV: 3B03.P00 VDEV: 3B03 Controller: CONTROL3 ACTIVE
Uplink Port Connection:
Queue: z/VM
TX Packets: 266                 Discarded: 0       Errors: 0
TX Bytes: 27531
Queue: High
TX Packets: 803                 Discarded: 0       Errors: 0
TX Bytes: 82709
Queue: Normal
TX Packets: 497                 Discarded: 0       Errors: 0
TX Bytes: 51882
Queue: Low
TX Packets: 0                   Discarded: 0       Errors: 0
TX Bytes: 0
Totals
RX Packets: 0                   Discarded: 574414  Errors: 0
TX Packets: 1566                Discarded: 0       Errors: 0
RX Bytes: 0                     TX Bytes: 162122
Device: 3B03 Unit: 000   Role: DATA   Port: 0001
Options: VLAN_ARP
Partner Switch Capabilities: Reflective Relay
RDEV: 1E26.P00 VDEV: 1E26 Controller: CONTROL3 BACKUP
Adapter Connections:
Adapter Owner: LINUX6   NIC: 7000.P00 Name: VSWITCH3   Type: QDIO
PQUplinkTX: High
Porttype: Trunk
RX Packets: 1534           Discarded: 0       Errors: 0
TX Packets: 5             Discarded: 6       Errors: 0
RX Bytes: 159536          TX Bytes: 426
Device: 7002 Unit: 002   Role: DATA   Port: 0069
VLAN: 0007
Options: Broadcast Multicast IPv6 IPv4 VLAN
Unicast IP Addresses:
10.60.199.66             MAC: 02-22-33-00-00-52
50C0:C2C1::222:3300:100:52 MAC: 02-22-33-00-00-52 Local
FE80::222:3300:100:52 MAC: 02-22-33-00-00-52 Local
Multicast IP Addresses:
224.0.0.1               MAC: 01-00-5E-00-00-01
FF02::1                 MAC: 33-33-00-00-00-01
FF02::1:FF00:52        MAC: 33-33-FF-00-00-52 Local
Adapter Owner: TCPIP88   NIC: 8800.P00 Name: TCPIP88   Type: QDIO
PQUplinkTX: Normal
Porttype: Access
RX Packets: 776           Discarded: 0       Errors: 0
TX Packets: 780           Discarded: 0       Errors: 0
RX Bytes: 80610          TX Bytes: 80744
Device: 8800 Unit: 000   Role: DATA   Port: 0068
VLAN: IPv4 0007 Assigned by system
VLAN: IPv6 0007 Assigned by system
Options: Broadcast Multicast IPv6 IPv4 VLAN
Unicast IP Addresses:
```

```

10.60.199.65          MAC: 02-22-33-00-00-1E
50C0:C2C1::10:60:199:65 MAC: 02-22-33-00-00-1E Local
50C0:C2C1::10:60:199:94 MAC: 02-22-33-00-00-1E Local
FE80::222:3300:100:1E MAC: 02-22-33-00-00-1E Local
Multicast IP Addresses:
224.0.0.1            MAC: 01-00-5E-00-00-01
FF02::1              MAC: 33-33-00-00-00-01 Local
FF02::2              MAC: 33-33-00-00-00-02 Local
FF02::1:FF00:1E     MAC: 33-33-FF-00-00-1E Local
FF02::1:FF99:65     MAC: 33-33-FF-99-00-65 Local
FF02::1:FF99:94     MAC: 33-33-FF-99-00-94 Local
Adapter Owner: TCPIP89 NIC: 8900.P00 Name: TCPIP89 Type: QDIO
PQUplinkTX: Normal
Porttype: Access
RX Packets: 770      Discarded: 0      Errors: 0
TX Packets: 781      Discarded: 0      Errors: 0
RX Bytes: 80050      TX Bytes: 80952
Device: 8900 Unit: 000 Role: DATA      Port: 0066
VLAN: IPv4 0007 Assigned by system
VLAN: IPv6 0007 Assigned by system
Options: Broadcast Multicast IPv6 IPv4 VLAN
Unicast IP Addresses:
10.60.199.67        MAC: 02-22-33-00-00-1B
50C0:C2C1::10:60:199:67 MAC: 02-22-33-00-00-1B Local
FE80::222:3300:100:1B MAC: 02-22-33-00-00-1B Local
Multicast IP Addresses:
224.0.0.1            MAC: 01-00-5E-00-00-01
FF02::1              MAC: 33-33-00-00-00-01 Local
FF02::2              MAC: 33-33-00-00-00-02 Local
FF02::1:FF00:1B     MAC: 33-33-FF-00-00-1B Local
FF02::1:FF99:67     MAC: 33-33-FF-99-00-67 Local

```

### 3. QUERY VSWITCH PROMiscuous:

This example shows the users who are authorized to use promiscuous mode on the simulated network adapter.

```

cp q vswitch wadp pro
VSWITCH SYSTEM WADP      Type: VSWITCH Connected: 0      Maxconn: INFINITE
PERSISTENT RESTRICTED  NONROUTER      Accounting: OFF
USERBASED LOCAL
VLAN Aware Default VLAN: 0001      Default Porttype: Trunk      GVRP: Disabled
Native VLAN: 0001      VLAN Counters: Off
MAC address: 02-6F-5C-00-00-1E      MAC protection: ON
IPTimeout: 5      QueueStorage: 8
Isolation Status: OFF      VEPA Status: OFF
Unicast IP address count: 0
Promiscuous ports:
2176 Userid: TCPIPX
2177 Userid: TCPIP1
Uplink port:
State: Defined      PriQueueing: OFF
PMTUD setting: EXTERNAL      PMTUD value: 65535      Trace Pages: 8
RDEV: 1000.P00      Controller: NONE      Error: No RDEV

```

### 4. QUERY VSWITCH PORTNUMBER:

This example shows the ports that are authorized to connect to the virtual switch with a simulated network adapter.

```

cp q vswitch vswpb3 portnumber
VSWITCH SYSTEM VSWPB3  Type: QDIO      Connected: 0      Maxconn: INFINITE
PERSISTENT RESTRICTED  IP      Accounting: OFF
PORTBASED LOCAL
VLAN Aware Default VLAN: 0008      Default Porttype: Access      GVRP: Disabled
Native VLAN: 0001      VLAN Counters: OFF
MAC address: 02-00-22-00-00-09      MAC Protection: Unspecified
State: Defined
IPTimeout: 5      QueueStorage: 8
Isolation Status: OFF      VEPA Status: OFF
Unicast IP address count: 0
Ports:
0010 Porttype: Access Userid: SUSAN H      VLAN: 0008
0011 Porttype: Access Userid: JOE N      VLAN: 0008
0012 Porttype: Access Userid: MARY N      VLAN: 0008
0100 Porttype: Trunk Userid: SAM N      VLAN: 0002 0003 0008
VLAN: 0009-0015 0047 0049
0101 Porttype: Trunk Userid: JULIA L      VLAN: 0010-0012

```

## QUERY VSWITCH

```
Uplink port:
State: Defined          PriQueuing: ON
PMTUD setting: EXTERNAL PMTUD value: 65535  Trace Pages: 8
```

### 5. QUERY VSWITCH DETAILS:

This is an example of a virtual switch that has an active Bridge Port.

```
cp q vswitch suef det
VSWITCH SYSTEM SUEF      Type: QDIO    Connected: 2    Maxconn: INFINITE
PERSISTENT RESTRICTED  ETHERNET      Accounting: OFF
USERBASED LOCAL
VLAN Unaware
MAC address: 02-00-00-00-00-A7  MAC Protection: Unspecified
IPTimeout: 5      QueueStorage: 8
Isolation Status: OFF      VEPA Status: OFF
Unicast IP address count: 2
Uplink Port:
State: Ready          PriQueuing: ON
PMTUD setting: EXTERNAL PMTUD value: 8992  Trace Pages: 8
RDEV: 4120.P00 VDEV: 4120 Controller: DTCVSW1 ACTIVE
Uplink Port Connection:
Queue: z/VM
TX Packets: 5          Discarded: 0      Errors: 0
                          TX Bytes: 27531
Queue: High
TX Packets: 0          Discarded: 0      Errors: 0
                          TX Bytes: 82709
Queue: Normal
TX Packets: 7          Discarded: 0      Errors: 0
                          TX Bytes: 51882
Queue: Low
TX Packets: 0          Discarded: 0      Errors: 0
                          TX Bytes: 0

Totals:
RX Packets: 0          Discarded: 0      Errors: 0
TX Packets: 12         Discarded: 0      Errors: 0
RX Bytes: 0            TX Bytes: 1128
Device: 4120 Unit: 000 Role: DATA Port: 2049
Partner Switch Capabilities: Reflective Relay
RDEV: 4123.P00 VDEV: 4123 Controller: DTCVSW1 BACKUP
Bridge Port:
Role: Secondary      Status: Active      Active LPAR: BCT2
State: Ready
RDEV: F200 VDEV: F200 Controller: DTCVSW1
Bridge Port Connection:
MFS: 16K Buffer Limit: 2048 In use: 0 Trace Pages: 8
NIC Distribution: ON
RX Packets: 0          Discarded: 0      Errors: 0
TX Packets: 0          Discarded: 0      Errors: 0
RX Bytes: 0            TX Bytes: 0
Asynchronous Requests: 0 Unavailable Buffers: 0
Device: F200 Unit: 000 Role: DATA Port: 2057
HyperSockets Bridge Capable Ports: Connected: 1
ID: LAPLACE3.F212
RX Packets: 0          Discarded: 0      Errors: 0
TX Packets: 0          Discarded: 0      Errors: 0
RX Bytes: 0            TX Bytes: 0
Options: Ethernet Broadcast
Unicast MAC Addresses:
06-00-F2-03-02-12 IP: 10.1.7.10
Multicast MAC Addresses:
01-00-5E-00-00-01
33-33-00-00-00-01
33-33-FF-03-02-12
Adapter Connections: Connected: 1
Adapter Owner: LAPLACE4 NIC: 0710.P00 Name: UNASSIGNED Type: QDIO
PQUplinkTX: Normal
RX Packets: 2482       Discarded: 0      Errors: 0
TX Packets: 6          Discarded: 0      Errors: 0
RX Bytes: 165819       TX Bytes: 468
Device: 0712 Unit: 002 Role: DATA Port: 0001
Options: Ethernet Broadcast
Unicast MAC Addresses:
02-00-00-00-00-A8 IP: 10.1.7.3
02-00-00-00-00-A8 IP: 50C0:C216::10:1:7:3
Multicast MAC Addresses:
01-00-5E-00-00-01
```

33-33-00-00-00-01  
33-33-FF-00-00-A8

## 6. QUERY VSWITCH IVLSW:

This is an example of an IVL virtual switch.

```

cp q vswitch ivlsw
VSWITCH SYSTEM IVLSW      Type: IVL      Domain: A      Maxconn: INFINITE
PERSISTENT RESTRICTED    ETHERNET      Accounting: OFF
PORTBASED  LOCAL
VLAN Unaware
MAC address: 02-12-34-00-00-08      MAC Protection: Unspecified
IPTimeout: 5      QueueStorage: 8
Isolation Status: OFF      VEPA Status: OFF
Unicast IP address count: 0
Uplink Port:
State: Ready      PriQueuing: ON
PMTUD setting: EXTERNAL      PMTUD value: 8992      Trace Pages: 8
RDEV: 7292.P00 VDEV: 7292      Controller: DTCVSW1 ACTIVE
Adapter ID: 29640007E386.0370
Uplink Port Connection:
Queue: z/VM
TX Packets: 13      Discarded: 0      Errors: 0
TX Bytes: 1520
Queue: High
TX Packets: 6      Discarded: 0      Errors: 0
TX Bytes: 520
Totals:
RX Packets: 88      Discarded:0      Errors: 0
TX Packets: 19      Discarded:0      Errors: 0
RX Bytes: 3696      TX Bytes: 2040
Partner Switch Capabilities: Unavailable
IVL Port:
Adapter Owner: SYSTEM      NIC: FFFD.P00 Name: UNASSIGNED Type: QDIO
RX Packets: 2462      Discarded: 0      Errors: 0
TX Packets: 17      Discarded: 0      Errors: 0
RX Bytes: 105852      TX Bytes: 854
Device: FFFD Unit: 000      Role: DATA      Port: 2100
Options: Ethernet Broadcast Multicast.
Unicast MAC Addresses:
02-12-34-00-00-00
Multicast MAC Addresses:
03-FF-FF-FF-FF-05

```

## 7. QUERY VSWITCH GVSW DETAILS:

This is an example of a global virtual switch.

```

cp q vswitch gvsw details
VSWITCH ASYSTEM.GVSW      Type: QDIO      Connected: 1      Maxconn: INFINITE
PERSISTENT RESTRICTED    ETHERNET      Accounting: OFF
USERBASED  GLOBAL
VLAN Unaware
MAC address: 02-12-34-00-00-07      MAC Protection: Unspecified
IPTimeout: 5      QueueStorage: 8
Isolation Status: OFF      VEPA Status: OFF
Unicast IP address count: 0
Uplink Port:
State: Ready      PriQueuing: OFF
PMTUD setting: EXTERNAL      PMTUD value: 65535      Trace Pages: 8
RDEV: 7230.P00 VDEV: 7230      Controller: DTCVSW1 ACTIVE
Adapter ID: 29640007E386.0251
Uplink Port Connection:
RX Packets: 0      Discarded: 0      Errors: 0
TX Packets: 202      Discarded: 0      Errors: 0
RX Bytes: 0      TX Bytes: 27884
Device: 7230 Unit: 000      Role: DATA      Port: 2049
Member: ASYSTEM.GVSW
State: Synchronized
Member: BSYSTEM.GVSW
State: Synchronized
Member: CSYSTEM.GVSW
State: Error

```

## Responses

Response 1:

```

VSWITCH switchname Type: type Connected: count or Domain: X Maxconn: max
persistence restriction routermode Accounting: setting
user-port-based scope
VLAN awareness Default VLAN: nnnn Default Porttype: type GVRP: status
Native VLAN: nnnn VLAN Counters: setting
IP address: ipaddr Management ID: stacknm
MAC address: macaddr setting
IPTimeout: nnn QueueStorage: nnn
Isolation Status: nnn VEPA Status: nnn
Unicast IP address count: nnn
Ports:
  portnum(1) Porttype: porttype Userid: user Priority VLAN: vlan(1)
                                                vlan(2) vlan(3)
                                                vlan(4) vlan(5)
                                                vlan(6) vlan(7)
  portnum(2) Porttype: porttype Userid: user Priority VLAN: vlan(1)
                                                vlan(2)...
Promiscuous ports:
  portnum(1) Userid: user
  portnum(2) Userid: user
Controller list:
  user(1) user(2) user(3) user(4) user(5) user(6)
  user(7) user(8)
Uplink port:
State: setting PriQueuing: setting
PMTUD setting: setting PMTUD value: value Trace Pages: nnnn
Group: groupname active state LACP Mode: mode
RDEV: nnnn.Pnn VDEV: nnnn Controller: userid ACTIVE
NIC: userID VDEV: vdev
EQID: eqid
Adapter ID: adapterid
Uplink Port Connection:
Data Counters Cleared: mm/dd/yy hh:mm:ss
Discarded Counters Cleared: mm/dd/yy hh:mm:ss
Error Counters Cleared: mm/dd/yy hh:mm:ss
MAC address: macaddr
Queue: z/VM
  TX Packets: txpackets Discarded: nnn Errors: nnn
  TX Bytes: txbytes
Queue: High
  TX Packets: txpackets Discarded: nnn Errors: nnn
  TX Bytes: txbytes
Queue: Normal
  TX Packets: txpackets Discarded: nnn Errors: nnn
  TX Bytes: txbytes
Queue: Low
  TX Packets: txpackets Discarded: nnn Errors: nnn
  TX Bytes: txbytes
Totals:
RX packets: rxpackets Discarded: nnn Errors: nnn
TX packets: txpackets Discarded: nnn Errors: nnn
RX bytes: rxbytes TX bytes: txbytes
Device: vdev Unit: nnn Role: role Port: portnum
Options: options Unicast IP Addresses:
ipaddr MAC: macaddr <flags>
Partner Switch Capabilities: options
Backup Devices:
RDEV: nnnn.Pnn VDEV: nnnn Controller: userid BACKUP
Bridge port:
Role: role Status: status Active LPAR: lparname
State: setting
RDEV: nnnn VDEV: nnnn Controller: userid
EQID: eqid
Bridge Port Connection:
Data Counters Cleared: mm/dd/yy hh:mm:ss
Discarded Counters Cleared: mm/dd/yy hh:mm:ss
Error Counters Cleared: mm/dd/yy hh:mm:ss
MFS: nnK Buffer Limit: nnn InUse: nnn Trace Pages: nnnn
NIC Distribution: setting
RX packets: rxpackets Discarded: nnn Errors: nnn
TX packets: txpackets Discarded: nnn Errors: nnn
RX bytes: rxbytes TX bytes: txbytes
Asynchronous Requests: nnn Unavailable Buffers: nnn
Device: vdev Unit: nnn Role: role Port: portnum

```

```

HyperSockets Bridge Capable Ports:    Connected: count3
ID: logicalGuestPortID
VLAN: vlan(1) vlan(2) vlan(3) vlan(4)
      vlan(5) vlan(6) vlan(7) vlan(8)
      vlan(9) ...                       vlan(n)
RX packets: rxpackets                  Discarded: nnn      Errors: nnn
TX packets: txpackets                  Discarded: nnn      Errors: nnn
RX bytes: rxbytes                       TX bytes: txbytes
Options: options
Unicast MAC Addresses:
  macaddr mac-status
Multicast MAC Addresses:
  macaddr
  macaddr
  macaddr
Member: systemname.switchname
State: state
Adapter Connections:                   Connected: count2
Adapter Owner: userid NIC: dev.Pnn Name: portname Type: type
PQUpLinkTX: priority
Porttype: porttype
Port: portnum
RX packets: rxpackets                  Discarded: nnn      Errors: nnn
TX packets: txpackets                  Discarded: nnn      Errors: nnn
RX bytes: rxbytes                       TX bytes: txbytes
Device: vdev Unit: nnn Role: role Port: portnum
VLAN: ipversion nnnn(1) status
Router: type ipversion vlan
Options: options
Unicast IP Addresses:
  ipaddr MAC: macaddr <flags>
  ipaddr MAC: macaddr <flags>
Multicast IP Addresses:
  mpaddr MAC: macaddr <flags>
  mpaddr MAC: macaddr <flags>
IVL Port:
Adapter Owner: userid NIC: dev.Pnn Name: portname Type: type
Porttype: porttype
RX packets: rxpackets                  Discarded: nnn      Errors: nnn
TX packets: txpackets                  Discarded: nnn      Errors: nnn
RX bytes: rxbytes                       TX bytes: txbytes
Device: vdev Unit: nn Role: role Port: portnum
VLAN: nnnn
Options: options
Unicast MAC Addresses:
  macaddr
  macaddr
Multicast MAC Addresses:
  macaddr
  macaddr

```

The fields in this response are described as follows:

**VSWITCH**

[SYSTEM *switchname* | *systemname.switchname*] uniquely identifies the virtual switch. For GLOBAL virtual switches, *systemname.switchname* is shown in place of SYSTEM *switchname* where *systemname* is the 8 byte EBCDIC z/VM system identifier.

**Type**

*type* is the VSWITCH TYPE or the adapter type (QDIO or IVL).

**Connected**

- *count1* indicates the sum of the number of coupled adapters and the number of guest ports.
- *count2* indicates the number of coupled adapters.
- *count3* indicates the number of HyperSockets Bridge Capable guest ports.

**Domain x**

where *x* is the IVL VSWITCH domain and *x* can be A to H.

**Maxconn**

*max* indicates the maximum number of connections. The value is always INFINITE for a virtual switch.

***persistence***

indicates whether the virtual switch is PERSISTENT or TRANSIENT. The value is always PERSISTENT for a virtual switch.

### ***restriction***

indicates whether the virtual switch is RESTRICTED or UNRESTRICTED. All z/VM virtual switches are RESTRICTED.

### ***routermode***

for a VSWITCH of type IP (with an OSA-Express device as the UPLINK port) indicates whether NONROUTER or PRIROUTER was set for the VSWITCH. Otherwise, *routermode* indicates the VSWITCH is of type IP or ETHERNET.

### ***Accounting setting***

indicates whether accounting records will or will not be generated for this virtual switch.

### **USERBASED**

### **PORTBASED**

indicates whether the virtual switch is defined as user based or port based. The type determines what rules are applied when relocating a guest in an SSI.

### **GLOBAL**

### **LOCAL**

indicates whether the virtual switch is a local virtual switch or a global virtual switch, which can span systems running z/VM.

### ***VLAN awareness***

indicates whether the virtual switch is VLAN-aware or VLAN-unaware.

### ***Default VLAN nnnn***

indicates the default VLAN ID setting for the virtual switch.

*nnnn* indicates the default VLAN ID to be associated with a port when no VLAN IDs are specified. This is displayed only for VLAN-aware virtual switches. NONE indicates no default VLAN ID is associated with guest ports coupled to the virtual switch.

### ***Native VLAN nnnn***

indicates the native VLAN ID setting for the virtual switch.

*nnnn* indicates the VLAN ID to be associated with untagged frames that are received and transmitted by the virtual switch. NONE indicates no native VLAN has been assigned.

### ***VLAN Counters setting***

indicates the level of detail that is maintained for the virtual switch network activity.

*Off* indicates that the detailed VLAN counters are not needed. That is, transmission counters are kept at the port level. Packet counters are kept for each port to show the number of frames received, transmitted or discarded.

*On* indicates that transmission counters are kept at the VLAN level. That is, for each port, packet counters are maintained to show the number of frames received, transmitted or discarded for each VLAN.

### ***Default Porttype type***

indicates the default porttype attribute for guests authorized for the virtual switch. This is only displayed for VLAN-aware virtual switches.

### ***GVRP status***

indicates whether or not GVRP is enabled for a VLAN AWARE virtual switch with the GVRP attribute.

### ***IP Address***

*ipaddr* is the identity of the virtual switch. This is known as the management interface. *Ipaddr* is used by clients involved in switch management. For example, a Network Management Systems (NMS) requests Bridge MIB (Management Information Base) information using *ipaddr*.

### ***Management ID***

*stacknm* identifies the user ID of the TCP/IP stack providing switch management services.

### ***MAC address***

is the MAC address associated with the virtual switch.



**MAC protection**

*setting* indicates the MAC address protection level of protection to be used for network devices coupled to the virtual switch whose MAC address protection settings are not specified (set to UNSPECIFIED) by the CP SET NIC command.

**IPTimeout**

indicates the length of time in minutes that a transient IP address table entry remains in the IP address table for the virtual network. A transient entry is an entry added to the virtual network by one guest on behalf of another. This value has no meaning for ETHERNET networks.

**QueueStorage**

*nnn* is the maximum amount of fixed storage that CP will allocate for QDIO queues used for data transfer.

**Isolation Status**

*nnn* indicates the isolation status of the VSWITCH and its UPLINK port data connection.

**VEPA Status**

*nnn* indicates the VEPA status of the VSWITCH and its UPLINK port data connection.

**Unicast IP address count**

*nnn* indicates the number of Unicast IP addresses associated with this VSwitch.

***user(1)***

represents one user ID in a list of users authorized to COUPLE to this virtual switch.

***vlan(1)***

indicates the list of VLAN IDs this user ID is authorized to access. *vlan(n)* may be ANY, indicating all VLAN IDs are authorized.

**State**

indicates the current state of the virtual switch. For more information about the states, see [“Usage Notes” on page 1352](#).

**PriQueuing**

indicates the current priority queuing setting for the virtual switch uplink port. The values for *setting* are: ON, OFF, and FORCED OFF.

**ON**

The virtual switch will exploit priority queuing. If the customer configured the OSA-Express features used by a virtual switch uplink port for priority queuing via IOCP (PQ\_ON), then z/VM will establish one input queue and four output queues when activating its network connection. This will allow z/VM to transmit data to the external network at four different priorities. CP will use the highest priority queue for control and management traffic. The other three queues (low, normal and high) can be used for virtual NIC's network connections.

**OFF**

The virtual switch will not exploit priority queuing. All outbound data to the external network is transmitted with equal priority.

**FORCED OFF**

The IVL virtual switch is configured to exploit priority queuing but priority queuing is forced off by z/VM because the OSA-Express features used by an IVL virtual switch uplink port are configured without priority queuing (PQ\_OFF).

The default for TYPE IVL virtual switches is PRIQUEUING ON, given priority queuing of both management and encapsulated production data is required to maintain a viable IVL Network. Only an OSA-Express feature configured for priority queuing should be used on an IVL virtual switch's uplink port.

**PMTUD setting *setting***

indicates the current Path MTU Discovery setting for the Uplink RDEV port device, configured either by the SET VSWITCH UPLINK PATHMTUDISCOVERY command or assigned as the default. The values for *setting* are: EXTERNAL (the default), Value, or Off.

**Note:** If the virtual switch is defined with either the NOUPLINK or UPLINK NIC options, the PMTUD setting and value fields are not displayed.

**PMTUD value *value***

indicates the Path MTU Discovery value, *value*, currently in effect for the virtual switch Uplink Port.

**Note:** This field is not displayed if the Path MTU Discovery setting is Off.

**Group *groupname***

is the port group assigned to this virtual switch, if any.

**LACP Mode *mode***

indicates whether LACP is Active or Inactive for the port group.

**RDEV**

*nnnn.Pnn* indicates the base real device number and the OSA-Express port number of the QDIO device attached to the virtual switch or specified as an alternate real device address. If the value "Error" is displayed on the same line, the device is not functional, and the problem must be corrected. Where possible, the error condition is included. Examine the operator console and the console of the z/VM TCP/IP stack controller for more information about the error.

Error conditions for both the Uplink and Bridge Port devices include:

- Operational
- Portname conflict
- No Layer 2
- No RDEV
- RDEV attached
- Invalid port number
- Incompatible for VSWITCH type
- Initialization
- Stalled OSA-E
- Stalled controller
- Path severed
- Routing conflict
- Offline
- Detached
- Type mismatch
- TCP/IP conflict
- Invalid device state
- Not ready
- TCP/IP resources
- No Link Aggregate support
- Hardware attributes
- Exclusive use error
- Unexpected internal error
- Attempting restart
- No OSA QDIO Connection Isolation
- Incompatible controller
- EQID mismatch
- No HiperSockets Bridge support
- HiperSockets Bridge error
- No VEPA support
- No Reflective Relay

- Reflective Relay error
- SET VSWITCH SWITCHOVER issued
- No priority queuing

**VDEV**

*nnnn* indicates the base virtual number used by the virtual switch controller for this RDEV.

**Controller**

is the user ID of the TCP/IP virtual machine acting as a controller for this RDEV.

**NIC**

*userID vdev* indicates the virtual device address of the NIC attached to the VSWITCH as the UPLINK port. If the value "Error" is displayed next to the NIC address, the device cannot be attached, and the problem must be corrected. Where possible, an explanation for the error is included. Error explanations include:

- No user
- Not authorized
- No VDEV
- VDEV attached
- Incompatible for VSWITCH type
- VLAN conflict
- Type mismatch
- No MAC address
- Port Error
- Unexpected error

**EQID *eqid***

indicates the EQID for the device, if any.

**Adapter ID *adapterid***

describes the OSA-Express port in use on system *systemname*. The format is:

cccccccccccc.pchid

**cccccccccccc**

represents the 12 bytes of text character and hexadecimal information that uniquely identifies the processor.

**pchid**

identifies the Physical Channel ID representing the OSA-Express adapter.

**ACTIVE**

indicates the port is currently available and in use.

**BACKUP**

indicates whether the alternate real device address has been initialized as a backup device to the base real device.

**Data Counters Cleared**

Indicates that the transmitted and received counters were reset to zero via the SET VSWITCH COUNTERS CLEAR DATA command at the time stated in the message.

**Discarded Counters Clearer**

Indicates that the discarded counters were reset to zero via the SET VSWITCH COUNTERS CLEAR DISCARDED command at the time stated in the message.

**Error Counters Cleared**

Indicates that the errors counters were reset to zero via the SET VSWITCH COUNTERS CLEAR ERRORS command at the time stated in the message.

### **MAC address *macaddr***

is the real or virtual MAC address associated with the UPLINK port device. For an IP VSWITCH, this is the adapter MAC address. For an ETHERNET VSWITCH, this is a virtual MAC address.

### **RX packets**

*rxpackets* is the number of packets received. If "Data Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DATA command.

### **TX packets**

*txpackets* is the number of packets transmitted. If "Data Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DATA command.

### **Discarded**

*nnn* indicates the number of packets discarded. A packet can be discarded because it is not formatted correctly, for example, or contains an unauthorized VLAN ID. Packets can also be discarded if the virtual switch's internal buffer limit has been reached. The internal buffer limit is reached when the virtual switch is unable to send packets over the OSA port or deliver packets to a guest VM port. Examples of this include: (1) An OSA port (RDEV) output buffer is not being serviced quickly enough by the adapter. A fully utilized OSA adapter will cause this condition. (2) A guest VM port input buffer is not being serviced quickly enough by the guest operating system. This can happen when a highly utilized guest operating system needs additional resources (memory or processor) to handle the workload. If "Discarded Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DISCARDED command.

### **Errors**

*nnn* indicates the number of error packets discarded. If "Error Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR ERRORS command.

### **RX bytes**

*rxbytes* is the number of bytes received. If "Data Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DATA command.

### **TX bytes**

*txbytes* is the number of bytes transmitted. If "Data Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DATA command.

### **Bridge Port Role**

indicates the Bridge Port is configured for either the Primary or Secondary role.

### **Status *status***

indicates the status of the Bridge Port. It can be either Active, Inactive, Standby, or Unknown.

#### **Active**

currently provides the bridge connection to the HiperSockets CHPID.

#### **Inactive**

indicates the configured Bridge Port is disconnected and unable to assume the active Bridge Port connection.

#### **Standby**

indicates the Secondary Bridge Port connection is waiting to or available to become the active Bridge Port connection.

#### **Unknown**

indicates the Bridge Port connection is in transition. This status is temporary.

### **Active LPAR *lparname***

is acting as the Active Bridge Port connection, if known. Otherwise, Unknown is displayed.

### **State**

indicates the current state of the virtual switch's Bridge Port. See ["Usage Notes" on page 1352](#) for the list of possible state values and their descriptions.

**RDEV**

*nnnn* indicates the base real device number of the HiperSockets device used for the Bridge Port. If the value "Error" is displayed next to the RDEV address, the device cannot be attached and the problem must be corrected. Where possible, an explanation for the error is included. Examine the operator console and the console of the z/VM TCP/IP stack controller for more information about the error. See the error conditions listed under RDEV for a list of possible Bridge Port errors.

**VDEV**

*nnnn* indicates the base virtual number used by the virtual switch controller for this Bridge Port RDEV.

**Controller**

is the user ID of the TCP/IP virtual machine acting as a controller for this Bridge Port RDEV.

**Data Counters Cleared**

Indicates that the transmitted and received counters were reset to zero via the SET VSWITCH COUNTERS CLEAR DATA command at the time stated in the message.

**Discarded Counters Cleared**

Indicates that the discarded counters were reset to zero via the SET VSWITCH COUNTERS CLEAR DISCARDED command at the time stated in the message.

**Error Counters Cleared**

Indicates that the errors counters were reset to zero via the SET VSWITCH COUNTERS CLEAR ERRORS command at the time stated in the message.

**MFS**

*nnK* is the Maximum Frame Size configured for the HiperSockets CHPID based on the IOCP channel parameters. If the MFS cannot be determined (for example, if the device is not found), the MFS value will be displayed as "---".

**Buffer Limit**

indicates the buffer limit (BUFFERS) configured for the Bridge Port on the SET VSWITCH command or MODIFY VSWITCH statement.

**InUse**

indicates how many buffers configured are currently in use.

**Trace Pages**

indicates the number of pages allocated for use by the Bridge Port or UPLINK Port internal trace table. This size is set by the SET VSWITCH BRIDGEPORT TRACESIZE or SET VSWITCH UPLINK TRACESIZE command.

**NIC Distribution**

indicates the status of NIC distribution for the Bridge Port. The value can be either OFF (default) or ON. For more information, see the NICDISTRIBUTION operand of the ["SET VSWITCH"](#) on page 1757 command.

**RX packets**

*rxpackets* is the number of packets received on the Bridge Port. If "Data Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DATA command.

**TX packets**

*txpackets* is the number of packets transmitted on the Bridge Port. If "Data Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DATA command.

**Discarded**

*nnn* indicates the number of packets discarded on the Bridge Port. If "Discarded Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DISCARDED command.

**Errors**

*nnn* indicates the number of error packets discarded on the Bridge Port. If "Error Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR ERRORS command.

**RX bytes**

*rxbytes* is the number of bytes received on the Bridge Port. If "Data Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DATA command.

**TX bytes**

*txbytes* is the number of bytes transmitted on the Bridge Port. If "Data Counters Cleared:" is displayed above this value, the value was cleared at the time indicated by the SET VSWITCH COUNTERS CLEAR DATA command.

**Asynchronous Requests**

indicates the total number of completed asynchronous data transfers.

**Unavailable Buffers**

indicates the number of times the virtual switch or HiperSockets firmware could not perform an asynchronous request due to unavailable buffers on the target port.

**Device**

*vdev* indicates one of the devices assigned to this connection.

**Unit**

*nnn* indicates the relative device number of this adapter. The base device is unit 000.

**Role**

*role* indicates the role assigned to this device (CTL-READ, CTL-WRITE, or DATA).

**HiperSockets Bridge Capable Ports**

section describes the guests on the HiperSockets channel that are bridge capable.

**ID *logicalGuestPortID***

represents a specific bridge capable HiperSockets device. It will be displayed in one of the following formats:

- The HiperSockets device is attached to a z/VM guest on the same LPAR as the Bridge Port. The *logicalGuestPortID* will be displayed as a z/VM user ID followed by the device number. For example: LINUX001.1F00.
- The HiperSockets device is in use on a different LPAR and the LPAR name can be determined. The format will be an LPAR name followed by the device number. For example: SYSTEM1.304.
- The HiperSockets device is in use on a different LPAR and the LPAR name cannot be determined. z/VM generated a unique value that will be displayed followed by the device number. For example: LPAR010F.308.

**VLAN *vlan(1) vlan(2) ... vlan(n)***

indicates the list of VLAN IDs that are associated with the *logicalGuestPortID*.

**Options**

*options* indicates the currently enabled features for this *logicalGuestPortID*, such as Broadcast.

***macaddr***

indicates a unicast or multicast MAC address associated with the *logicalGuestPortID*.

***status Error***

indicates that this MAC address in the virtual switch LAN could not be registered. This could indicate a conflict for this particular MAC address.

**Member: *systemname.switchname***

describes the members of the global virtual switch where *systemname* identifies the 8 byte EBCDIC z/VM system identifier of the system where a member of the global virtual switch exists.

**State *state***

indicates the state of the global member.

**Synchronized**

the local system's view of the global virtual switch matches the configuration on other IVL domain members.

**Pending Synchronization**

the local system is checking to make sure that configuration of the global virtual switch matches the existing configuration on all systems in the IVL domain.

**Error [Reason:msgno]**

the local system has identified a configuration mismatch with the way the global virtual switch is configured on another system in the IVL domain.

A message number will also be displayed to identify a specific synchronization problem. For more information on the reason, see [z/VM: CP Messages and Codes](#).

**Not Synchronized**

the local system's view of the global virtual switch does not match the configuration on other IVL domain members.

**IVL Port**

section describes the IVL NIC port.

When a IVL virtual switch has been defined, and the system is a member of an IVL domain, the display includes information describing the IVL port.

When the IVL virtual switch has been defined, but there is a system configuration which prevents the z/VM image from becoming an active IVL member, the following line is displayed. For more information on the reason, see [z/VM: CP Messages and Codes](#).

```
IVL Domain: Isolated          Reason: HCPIVLxxxxE
```

**Adapter Owner**

*userid* indicates the userid of the virtual machine owning the simulated NIC.

**NIC**

*dev.Pnn* indicates the base virtual device address and port number of the simulated NIC.

**Name**

*portname* indicates the portname assigned to the virtual adapter. The value "UNASSIGNED" is displayed if your software does not assign a value.

**PQUplinkTx**

*priority* indicates the priority with which outbound traffic will be sent to the external network (Low, Normal, High) if the virtual NIC is connected to a virtual switch. The PQUPLINKTX value will be displayed for all non-IVL virtual switches regardless of the priority queuing setting (ON and OFF). The PQUPLINKTX value will be used when priority queuing is enabled on the virtual switch (PRIQueuing ON).

**Porttype**

*porttype* indicates the port type of the simulated NIC.

**Port**

*portnum* is the port number associated with the guest connection or the virtual switch network connection.

**Device**

*vdev* indicates one of the devices assigned to this connection.

**Unit**

*nnn* indicates the relative device number of this adapter. The base device is unit 000.

**Role**

*role* indicates the role assigned to this device (CTL-READ, CTL-WRITE, or DATA).

**VLAN**

*ipversion* can be IPv4 or IPv6 when a single VLAN is set per IP Protocol.

*nnnn(n)* indicates the effective VLAN ID or IDs for the device. Multiple VLAN ID lines may be shown. If no VLANs are configured for a VLAN aware virtual switch, NONE will be displayed.

*status* is "Assigned by system" or "Assigned by user". If the specified VLAN ID conflicts with the authorizations for this user, "Conflicting VLAN ID" is displayed. "Inactive" is displayed when no active

VLANs are configured - that means that all ingress and egress frames will be discarded until a VLAN is configured.

### Router

*type ipversion* indicates the adapter has configured itself as a router with type Primary, Secondary or Multicast in IP version IPv4 or IPv6. *vlan* is either VLAN followed by the VLAN ID, if it is a VLAN-specific router, or Global, if it is a global router. If it is a global router, lines will follow to describe which VLAN IDs the adapter is routing for.

The router line is displayed only when a router is configured.

### Options

*options* indicates the currently enabled features for this interface, such as Broadcast, Multicast, IPv4 or IPv6. A value of VLAN\_ARP means the controller can register IP addresses on the OSA-Express with the proper VLAN groups.

### Partner Switch Capabilities

*options* indicates the capabilities of the partner switch. If CP is unable to retrieve the capabilities (for example, because of lack of support in the OSA-Express device), "Unavailable" is displayed. Otherwise the response is "Reflective\_Relay" if the partner switch has that capability (required for VEPA) or "No Reflective\_Relay."

### Take-over Status Member: *systemname* MAC address: *mac*

an additional line(s) shows take over status when the shared port group is involved in MAC address takeover to maintain connectivity following a failure.

A different global VSwitch member had an UPLINK port failure, and its MAC addresses are now assigned to one of the local OSAs on this global VSwitch member. The member name may be omitted if it cannot be obtained (for example if the member left the IVL domain). This is followed by a list of MAC Addresses.

### *ipaddr*

represents one Unicast IP address that has been registered to this device and optional information relating to the IP address.

An IPv4 address is displayed in dotted-decimal format, for example 9 . 130 . 44 . 144.

An IPv6 address is displayed in compressed format (":" designates multiple 16-bit groups of zeros). For example, the IPv6 address

```
F802:0000:0000:0000:0000:0A00:0001
```

is displayed in compressed format as F802 : :A00:1.

### *mpaddr*

represents one Multicast IP Address.

For IPv4, the address is displayed in dotted-decimal format with the associated MAC address.

For IPv6, the address is displayed in compressed format (":" designates multiple 16-bit groups of zeros).

### MAC

*macaddr* represents a MAC address.

### *flags*

additional information that may be displayed with the IP address. These include:

- Local indicates that this address was registered in a virtual switch LAN, but is NOT currently registered in the external network. This may indicate the external interface is disconnected. It may also indicate that this address is only supported within the immediate network (as it is in the case of an IPv6 address within an IP Layer VSWITCH).
- Conflict indicates that this address was registered locally but cannot be used at all because of a conflict with the network external to the virtual switch LAN.



- Remote indicates that this address was registered automatically to represent a source address found in a datagram. This entry is automatically removed after the IP Timeout (IPTimeout) has expired.
- Manual indicates that this address was registered manually by an ARP cache control command from the driver. This entry will be automatically removed after the IP Timeout (IPTimeout) has expired, or it may be removed earlier by the driver.
- NotOwner means that the IP address was registered by the guest using an ARP Add Unconditional setting, but another guest on the virtual switch had already registered the IP address. If the owner of the IP address removes its registration, this guest can become the owner and receive incoming traffic for the IP address.

## Messages

- HCP2783E VSWITCH *switchname* does not exist
- HCP2783E VSWITCH SYSTEM *switchnm* of type *type* does not exist
- HCP2836E VLAN was not found
- HCP3185I The displayed response may be incomplete. The system is not joined with the IVL Domain.

## QUERY VTOD

Notes:

<sup>1</sup> You can specify the operands in any order.

<sup>2</sup> A class G only user must be coupled to *userid* to use this operand.

### Authorization

Privilege Class: A, B, G

### Purpose

QUERY VTOD allows the virtual machine operator to interrogate the difference between their virtual machine's TOD clock and the system (hardware) TOD clock. This command also allows virtual machine operators to determine the set of logged-on virtual machines with TOD clock settings synchronized with their own.

A privileged user (class A or class B) can see such information for any logged-on user ID, not just the user's own virtual machine. Similarly, a class G user can see such information for any user ID to which the user is coupled.

### Operands

#### FOR *userid*

allows a privilege class user A or B to display the virtual TOD clock setting for any logged-on user ID. A class G user can display the same information for any user ID to which the user is coupled.

#### SYNC

allows the issuer to display the set of logged-on virtual machines with TOD clock settings that are in synchronism with their own. However, if the system TOD clock is in sync with their own, then only "System TOD" is displayed.

### Usage Notes

1. If the QUERY VTOD command is executed without any options, the virtual time-of-day of the user ID entering the command is displayed.
2. Instead of displaying user IDs virtual TOD clock settings, the SYNC option can be used to just display the system TOD.

### Responses

Response 1:

Three lines of numeric information are displayed followed by a list of user IDs if the SYNC option was used and the system TOD clock is not in sync with their own. This sample illustrates both portions:

```
System TOD: yyyy-mm-dd hh:mm:ss (xxxxxxxxxxxxxxxxxxxx)
User TOD:   yyyy-mm-dd hh:mm:ss (xxxxxxxxxxxxxxxxxxxx)
Delta:      xxxxxxxxxxxxxxxxxxxx
Synchronized with:
USERABCD USEREFGH USER1234 USER2345 USER4321 USER5647 USER2213 USERFRED
USERSAM USERJAKE USERRICK
```

In this sample, xxxxxxxxxxxxxxxx is a 16-digit hexadecimal number, and the other outputs are date and time stamps.

### Examples

To display your own offset, enter: `query vtod`.

If you are a class G only user and are coupled to CFVM1, you can display CFVM1's offset by entering: `query vtod for cfvm1`.

To display your own offset in addition to everyone with whom you are synchronized, enter: `query vtod sync`.

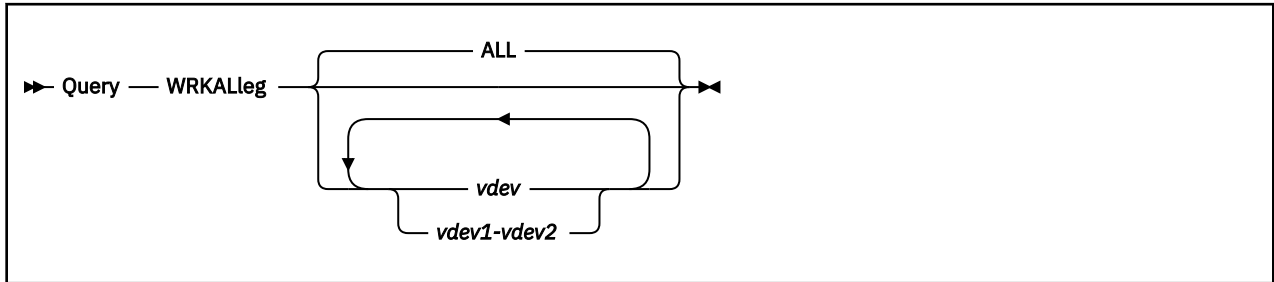
If you are a class G only user and are coupled to CFVM1, you can display CFVM1's offset and the names of everyone synchronized with CFVM1 by entering: `query vtod for cfvm1 sync`

If you are a class A or B user, you can display any VTOD offset in the system. For example, to display MAINT's VTOD setting and the set of users synchronized with MAINT, enter: `query vtod for maint sync`.

### Messages

- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP2800E *msgprocid* is not a CF service machine.
- HCP2805E One or more message processors does not exist in your I/O configuration.
- HCP2806E You are not allowed to control message processors.

## QUERY WRKALLEG



### Authorization

Privilege Class: G

### Purpose

Use QUERY WRKALLEG to display the status of working allegiance for each virtual device given. The device must be a minidisk.

### Operands

#### ALL

will display all the virtual devices in the user's configuration that have WRKALLEG ON.

#### vdev

#### vdev1-vdev2

is the virtual device number or a range of virtual device numbers of the minidisk.

### Usage Notes

1. If no virtual device is specified, all the virtual devices that are simulating working allegiance will be displayed.
2. VM supports a z/OS guest's participation in a Sysplex configuration only if all participating z/OS systems are guests of the same VM host. Environments which include a z/OS system running natively in a separate processor partition, or as a guest of a different VM system are not supported.
3. This command is rejected for full-pack minidisks on a shared real volume. Virtual working allegiance is incompatible with sharing real DASD. WRKALLEG is only for sharing virtual devices.
4. This command is rejected for dedicated DASD. Working allegiance is only necessary for shared minidisks. Virtual working allegiance is incompatible with sharing real DASD. WRKALLEG is only for sharing virtual devices.
5. Working allegiance applies only to users with write access to the minidisk. Furthermore, working allegiance is simulated only when a guest with write access initiates I/O. However, guests with read-only access will be allowed to query the device to determine if another guest issued SET WRKALLEG ON to the minidisk.
6. Working allegiance is always simulated for virtual disks in storage. QUERY WRKALLEG is rejected if the vdev is a virtual disk in storage.
7. The WRKALLEG setting applies to the minidisk, not to an individual virtual device. If two users have links to the same minidisk and one issues SET WRKALLEG, the setting will apply to both users' access to the minidisk.
8. For more information on working allegiance, see the principles of operation manual for your system.

## Responses

Response 1:

```
DASD <vdev> working allegiance is being
simulated among same system guests.
```

is returned for each virtual device specified where WRKALLEG=ON.

Response 2:

```
DASD <vdev> working allegiance is turned off.
```

is returned for each virtual device specified where WRKALLEG=OFF.

Response 3:

```
Working allegiance is being simulated for the
following minidisks:
0100 0200 0300 ...
```

is the response if QUERY WRKALLEG or QUERY WRKALLEG ALL is specified.

Response 4:

```
Working allegiance is not simulated for any
minidisk in your configuration.
```

is the response if QUERY WRKALLEG or QUERY WRKALLEG ALL is specified and no minidisks had WRKALLEG set ON.

## Messages

- HCP003E Invalid option - *{option|command contains extra option starting with option}*
- HCP006E Invalid device type - *vdev*
- HCP040E Device *vdev* does not exist
- HCP2010E DASD *vdev* is dedicated, WRKALLEG is rejected.
- HCP2010E DASD *vdev* is a full-pack mini on a shared real volume, WRKALLEG is rejected.
- HCP2010E DASD *vdev* is a VDISK, WRKALLEG is rejected.
- HCP6706E Invalid address range - *string*

## READY

---

```
▶▶ READY — vdev ▶▶
```

### Authorization

Privilege Class: G

### Purpose

Use READY to set pending a device-end interrupt for a specified virtual device.

### Operands

*vdev*

identifies the virtual device.

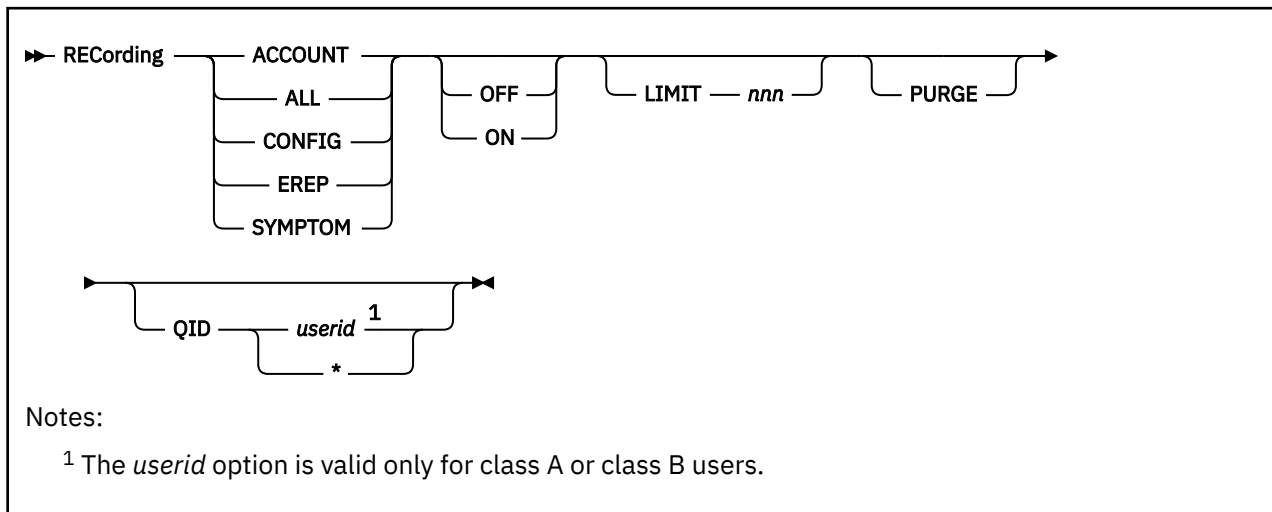
### Usage Notes

1. Use the READY command to clear a previous not-ready condition for a device.
2. Do not use the READY command for a virtual channel-to-channel adapter (CTCA). A not-valid device type error results.

### Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP263E Too many operands were supplied for this command.
- HCP1201E Device *vdev* cannot be readied because it is active.

## RECORDING



## Authorization

Privilege Class: A, B, C, E, F

## Purpose

Use RECORDING to change the processing parameters of the control program (CP) recording facilities.

## Operands

### ACCOUNT

#### ALL

#### CONFIG

#### EREP

#### SYMPTOM

specify the record type for which the parameter change is being requested. Note that the EREP parameter refers to LOGREC error records.

### ON

requests that recording of the specified record type or types be started. The association to the recording function will remain until a cold start of the system is performed.

### OFF

requests that recording of the specified record type or types be stopped.

### LIMIT *nnn*

specifies a decimal value, from 1 to 255, used as a threshold indicating when to notify the system operator and the receiving virtual machine that uncollected records are accumulating.

Uncollected records are records that have been created but not received by the recording virtual machine. Whenever the number of uncollected records of the specified type equals an integral multiple of LIMIT, the system operator and the virtual machine user who normally receives the records are notified.

For example, suppose you specify LIMIT 4. The system operator and the virtual machine user who normally receives the records are notified when four uncollected records accumulate. Notifications are issued when the number of uncollected records reaches 8, 12, 16, and so forth.

The default limit for ACCOUNT records is 20, for EREP records 2, for SYMPTOM records 2, and for CONFIG records 255. Any LIMIT setting that you indicate lasts until VM is restarted with COLD option.

**PURGE**

requests the purging of any records in storage that have been collected but not retrieved.

**QID *userid*****QID \***

identifies the user eligible to retrieve the records of the type specified for which CP's recording queue parameters are to be altered. If you are specifying your own user ID, you can specify *userid* as an asterisk (\*). You may specify your own user ID or asterisk (\*) to limit data selection to your own account.

As part of the z/VM system generation process, your system programmer may have indicated which virtual machines would retrieve records of a designated type. CP recording system services recognize additional virtual machines when the virtual machines connect to them using IUCV facilities. (The RETRIEVE CP utility processing connects to CP system services in order to collect CP records). Once a virtual machine has connected to a CP recording system service, its recording queue is maintained by CP until z/VM is restarted with the COLD option. If the user ID you specify with the QID operand is different from the user ID your system programmer specified at system generation time and the user ID has never connected to the appropriate CP system service with the RETRIEVE utility, the change is not known to the CP recording system service and the user ID will not appear in the query recording output.

**Note:** The QID operand is valid only for class A and B users.

**Usage Notes**

1. If you are a class A or B user and you omit *userid*, the command applies to all records being collected by CP that match the specified record types.
2. If you do not specify any of the optional operands, ON is assumed.
3. Specifying the OFF operand prevents newly generated records from being queued for the relevant virtual machines, but leaves all previously generated records queued. Leaving a large number of these records queued in CP storage could affect system performance and the availability of CP storage. You can purge these records using the PURGE operand (previously described). Also, see [Stopping and Starting the Recording of Accounting Records in z/VM: System Operation](#).
4. If a user has recording set OFF and a non-zero number of records queued, recording cannot be turned ON for that user until the queued records are processed. CP issues message HCP8087I if you try to turn recording on while records are queued. You can use the PURGE operand (previously described) to purge the records, or the RETRIEVE utility to retrieve the records from the queue and then you can turn recording on.

**Responses**

Response 1:

```
COMMAND COMPLETE
```

indicates that the command has completed successfully.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]



- HCP6525E Authorization request failed.
- HCP8058I User *userid* has purged *count* records from the {\*LOGREC|\*ACCOUNT|\*SYMPTOM| \*CONFIG} recording queue.
- HCP8072I Recording entry not found
- HCP8087I Records are queued for user *userid* on the {\*LOGREC|\*ACCOUNT|\*SYMPTOM| \*CONFIG} recording queue and must be purged or retrieved before recording can be turned on.

## REDEFINE



### Authorization

Privilege Class: G

### Purpose

Use REDEFINE to change the virtual device number of any existing virtual device in your configuration.

### Operands

#### **vdev1**

is the virtual device number of an existing virtual device.

#### **vdev2**

is the new virtual device number.

### Usage Notes

1. Virtual paging subsystem devices should not be redefined in order to protect the integrity of the user's virtual paging subsystem.
2. When you use the REDEFINE command to alter the configuration of your virtual machine, the changes are temporary and are only in effect for the current session.
3. The virtual device numbers you specify can be no higher than FFFF.
4. Informational responses are sent to your display to confirm that the configuration change you requested has taken place. These responses do not appear if you have issued the SET IMMSG OFF command.
5. If you change the virtual device number of your console while spooling your console, the spool file device number will also change.
6. If the existing virtual device is part of a Concurrent Copy (CONCOPY) session for your virtual machine, the CONCOPY session is terminated in the storage control associated with the device.

### Responses

Response 1:

```
type vdev DEFINED
```

confirms that the device number has been changed.

<b>Type</b>	<b>Meaning</b>
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Line printer
PUN	Card punch

<b><u>Type</u></b>	<b><u>Meaning</u></b>
GRAF	Display device
CONS	Console
CTCA	Channel-to-channel adapter
CTLR	3745 Communications Controller
MSC	Mass Storage Controller
SWCH	Dynamic switching device
DEV	Any other device.

Response 2:

```

Redefining vdev1 has caused the termination of
CONCOPY session(s) on device(s): vdev...

```

indicates that Concurrent Copy (CONCOPY) sessions have been terminated during the process of changing the existing virtual device number.

***vdev1***

is the existing virtual device number that you are changing.

***vdev...***

is a list of the virtual device numbers that are part of the terminated CONCOPY sessions.

## Messages

- HCP002E Invalid operand - *operand*
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP092E Device *vdev* not defined; *device vdev* already defined
- HCP1150E DASD {*rdev/vdev*} is not a valid base exposure.
- HCP1200E DASD *vdev* has not been redefined because paging subsystem device cannot be redefined.

## REFRESH



### Authorization

Privilege Class: A

### Purpose

Use REFRESH to replace the existing logo configuration file maintained by CP or to tell CP to send new LOGO information to one or more VTAM service machines (VSM).

### Operands

#### LOGOinfo *fn ft*

tells CP to replace the existing logo configuration file with the file specified by *fn* and *ft*. The file must reside on a CMS-formatted minidisk accessed by CP.

#### LOGOVSM *userid*

sends new logo information to one or more VTAM service machines (VSM) specified by *userid*. *userid* is the user ID (or user ID pattern) of the VTAM service machine to which you want to send the new LOGO information. CP supports pattern matching as described in Usage Note [“4” on page 1380](#).

### Usage Notes

1. If CP encounters errors while reading the specified logo configuration file, CP displays an error message on the terminal of the user who issued the REFRESH LOGOINFO command. If errors are found, CP ignores the REFRESH LOGOINFO command.
2. Be careful when you change status area fields on a running system, because certain applications may depend on the status fields remaining constant.
3. If the file you requested does not reside on a CP-accessed minidisk, CP does not delete the previous logo configuration information.
4. Pattern matching follows these rules. Use an asterisk (\*) to match any number of characters. Use a percent symbol (%) to match any single character.
5. For more information on the contents of [The Logo Configuration File](#), see *z/VM: CP Planning and Administration*.
6. For information on CP-accessed minidisks, see [“CPACCESS” on page 89](#) and [“QUERY CPDISKS” on page 824](#).

### Responses

Response 1:

If the command is successful, CP does not issue a response.

Response 2:

If you enter a REFRESH LOGOINFO command and there are errors in the specified logo configuration file, you see error messages. Each error message is preceded by another message that tells you where the error occurs in the file.

Response 3:

If you enter a REFRESH LOGOVSM command and the VTAM service machine you requested is not found, you receive message HCP6784E.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - statement contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP6704E Missing token at end of line
- HCP6706E Invalid {fn|ft} - *fn|ft*
- HCP6737E Cannot find logo configuration file *fn ft*.
- HCP6784E No VSMS logged on
- HCP6784E No VSMS matching pattern *userid* logged on
- HCP6784E VSM *userid* not logged on
- HCP6787E No VSMS matching pattern *userid\_pattern* support dynamic logo updates.
- HCP6787E No VSMS support dynamic logo updates.
- HCP6787E VSM *userid* does not support dynamic logo updates.

## RELSPACE

```
▶▶ RELSPACE — vdev — ▶▶
```

### Authorization

Privilege Class: G

### Purpose

Use RELSPACE to return all extents on a device to its associated Extent Space Efficient pool. A user can release the space of any virtual device to which they have write access. This could be a dedicated DASD or a minidisk on a volume attached to the system.

### Operands

#### *vdev*

is the virtual device number of the device to have backing tracks released.

### Usage Notes

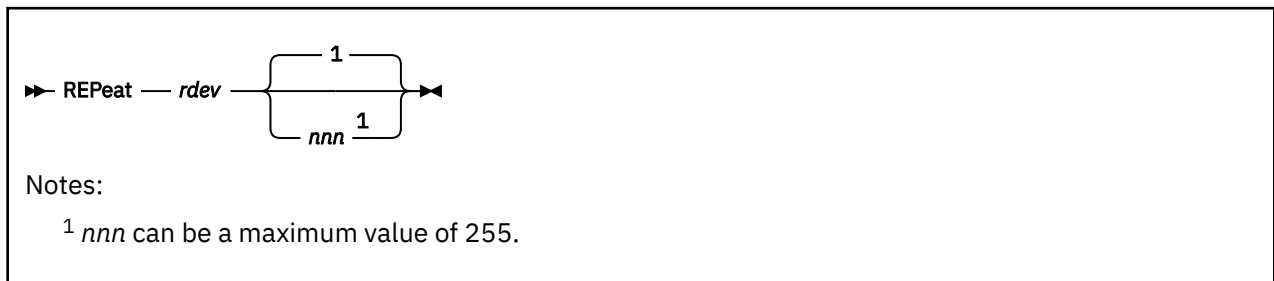
1. The RELSPACE command is intended for the release of extents on Extent Space Efficient (ESE) DASD volumes. RELSPACE can be used to return extents to the device's associated ESE pool to be used for expanded usage by other devices.
2. There is no guarantee that all data associated with the released tracks will be erased. If erasure is necessary, a policy such as using the CPFMTXA ERASE function before RELSPACE should be employed.
3. The Query DASD SPACE-EFFICIENT command can be used to check the ESE pool utilization for a given device and its associated pool.
4. Query DASD SPACE-EFFICIENT output will not correlate directly with the release of space of non-fullpack minidisks, when comparing the Query output before and after a release. The Release Allocated Space hardware function, invoked by the RELSPACE command, only releases FULL extents which begin and end on an integral hardware Extent boundary (for example, 21 cylinders). Since a non-fullpack minidisk can start and end on any real cylinder boundary, there may be partial hardware extents at the front and end of the minidisk which will not be released. Minidisks have to be defined to start and end on the real hardware Extent boundaries to circumvent this hardware restriction.
5. RELSPACE to non-fullpack minidisks, including real cylinder 1-to-end minidisks, will fail with I/O errors if issued to Extent Space Efficient volumes with only Phase 1 (Full Volume) Release Allocated Space hardware support. Phase 2 (Partial Volume Extent) or higher Release Allocated Space hardware support is required.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option* command contains extra option(s) starting with *option}*
- HCP006E Invalid device type - *rdev*
- HCP022E A virtual device number was not specified or it is invalid.
- HCP026E Operand was missing or invalid
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP263E Too many operands were supplied for this command.
- HCP6432I *vdev* is not on an Extent Space Efficient Volume and is not valid for the RELSPACE command.
- HCP6432I *vdev* is a PAV or HyperPAV alias device and is not valid for the RELSPACE command.

- HCP6432I *vdev* does not have write access and is not valid for the RELSPACE command.

## REPEAT



### Authorization

Privilege Class: D

### Purpose

Use REPEAT to increase the number of output file copies beyond the quantity originally requested.

### Operands

#### *rdev*

identifies the real output device (printer or punch) whose current active spool file is to be repeated.

#### **1**

#### *nnn*

identifies the number of copies to be made. If you do not specify a value, the number of extra copies defaults to 1; the maximum value you may specify for *nnn* is 255.

### Usage Notes

1. This command applies only to files that are currently being written to a real punch or printer.

### Responses

Response 1:

```
{PRT} rdev REPEATED  userid  FILE spoolid
{PUN}

RECS {nnnn} COPY [*] nnn SEQ sss CL c typ
     {nnnK}
     {nnnM}
```

#### *rdev*

identifies the real device whose current output is being repeated.

#### *userid*

identifies the user to receive the output.

#### *spoolid*

identifies the system-assigned file spoolid.

#### **RECS *nnnn***

#### **RECS *nnnK***

#### **RECS *nnnM***

identifies the number of spool file records remaining to be printed. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000



records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

**COPY [\*] *nnn***

identifies the number of copies remaining to be created.

When the optional asterisk (\*) is specified, the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

**SEQ *sss***

identifies the (printer or punch) file sequence number.

**CL *c***

is the spool file class.

***typ***

shows the device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP, or SYS.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP088E An invalid number was supplied for {COPY|FLASH} - *operand*.
- HCP140E *type {rdev/ldev}* attached to *userid*
- HCP141E *type rdev* not active
- HCP417E {*rdev/vdev*} is not an output spooling device.
- HCP418E {*rdev/vdev*} is not a spooled device.
- HCP6005E Option *option* is not supported by z/VM.

## REQUEST

---

► REQUEST ◄

### Authorization

Privilege Class: G

### Purpose

Use REQUEST to make pending an attention interrupt for your virtual console.

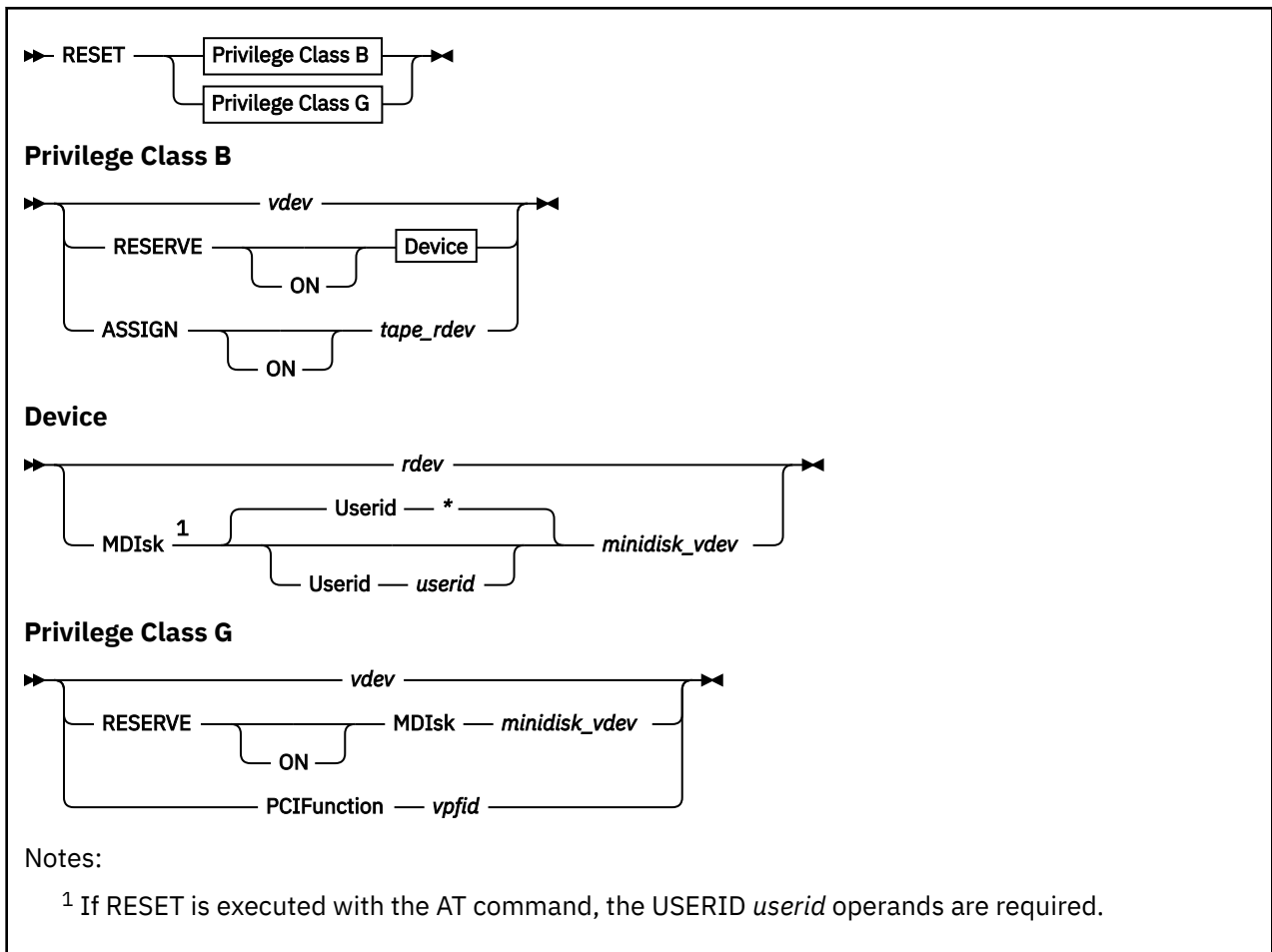
### Usage Notes

1. The REQUEST command performs the same functions as the ATTN command; you can use the two interchangeably.
2. The BEGIN command is not required after you enter REQUEST.
3. CP ignores the ATTN and REQUEST command if another attention interrupt is pending for your virtual machine, or if your virtual machine is not enabled for interrupts.

### Messages

- HCP002E Invalid operand - *operand*

## RESET



### Authorization

Privilege Class: B, G

### Purpose

Use RESET to clear all pending interrupts from a specified device.

A class B user can reset:

- The user's own virtual device to clear all pending interrupts.
- RESERVE on a real device or on another user's minidisk.
- ASSIGN on a real device.

A class G user can reset:

- The user's own virtual device to clear all pending interrupts.
- RESERVE on the user's own minidisk.
- A virtual PCI function to remove the function from the permanent error state.

## RESET

### Operands

#### *vdev*

is the virtual device number of the device to be reset.

#### **RESERVE**

releases the RESERVE status of a system shared DASD.

#### *rdev*

is the real device number of the device to be released.

#### **MDisk** *minidisk\_vdev*

is the virtual device number of the minidisk to be released.

#### **Userid \***

#### **Userid** *userid*

is the user ID of the user that has the *vdev*.

#### **ASSIGN**

unassigns a tape drive.

#### *tape\_rdev*

is the real device number of the tape drive to be released.


#### **PCIFunction**

resets a virtual (guest-attached) PCI function.

#### *vpfid*

specifies the virtual PCI function ID to be reset. Specify a virtual PCI function number as 8 hexadecimal digits. You can omit leading zeros.


### Usage Notes

1.  **Attention:** If you must release a device that is being reserved by CP for some user, you can use the RESET RESERVE command. You should use this command only when you know that the user for whom CP is holding the reserve is hung or is holding the reserve for too long and preventing access to the device by other users.
2. In addition to clearing all pending interrupts, all virtual sense bytes and error conditions occurring as a result of unit checks are reset.
3. The operator of a virtual system can issue the RESET command to return to CP any local displays that were connected to the virtual system using the DIAL command.
4. The reserve status of system DASD can be determined by issuing the Class B QUERY DASD RESERVE command.
5. The reserve status of an MDISK *vdev* can be determined by issuing the QUERY MDISK command.
6. When RESET RESERVE is issued to a device, that device immediately becomes available to the other sharing system.
7. If the specified tape drive is in a GIVEN state the issuer must be the user ID to whom the tape drive was given, or the command is not performed.
8. The RESET RESERVE command can, in some cases, be used to reset a device that is RESERVED BY SYSTEM. Normally a device in this state is in transition, and a subsequent query will indicate the device is reserved by a user, or not reserved. However, if querying the device consistently returns RESERVED BY SYSTEM, and other native systems seem to be unable to access the device, then use the RESET RESERVE command to release the outstanding reserve.
9. If a RESET *vdev* command is issued by a class B user using the AT command, the following message will be issued:

#### **HCP040E**

Device *vdev* does not exist

This is because virtual device addresses are not valid if the RESET command is issued using the AT command.

10.  **Attention:** The RESET ASSIGN command is intended for use in the removal of spurious assignments from a real device. You should use this command only when you know that the virtual machine user of the device, if any, is hung or has otherwise failed, and is thus unable to complete the work being done on the device. Use of this command on a device being actively used can allow the device to be accessed by another real system having shared access to the device, potentially destroying the active work.
11. To reset a real PCI function, use [“SET PCIFUNCTION” on page 1600](#) with privilege class B.

## Responses

Response 1:

```
DEVICE RESET - type vdev
```

is returned when you enter RESET *vdev*.

<b>Type</b>	<b>Meaning</b>
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Display device
CONS	Console
CTCA	Channel-to-channel adapter
CTLR	3745 Communications Controller
MSC	Mass Storage Controller
SWCH	Dynamic switching device
DEV	Any other device.

### ***type***

indicates the type of device that has been reset.

### ***vdev***

is the virtual device number.

Response 2:

```
RESERVE ON type rdev RESET
```

is returned when you enter the RESET RESERVE *rdev* command.

Response 3:

```
ASSIGN ON type rdev RESET
```

is returned when you enter the RESET ASSIGN *rdev* command.

Response 4:

```
RESERVE ON userid vdev RESET
```

is returned when you enter the RESET RESERVE *vdev* command.

Response 5:

## RESET

PCI function reset - *vpfid*

is returned when you enter the RESET PCIFUNCTION *vpfid* command.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option}* command contains extra option(s) starting with *option*}
- HCP006E Invalid device type - *vdev*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP022E A virtual PCI function ID was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP040E PCI function *vpfid* does not exist
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command.
- HCP1122E *command* was not performed. There is a return pending for device *vdev* from user *userid*.
- HCP1122E *command* was not performed. Tape *vdev* has been given to user *userid*.
- HCP1999E The assign was not reset on *rdev* because *{it is not a tape that supports assign|a time\_out condition occurred}*
- HCP2000E The reserve was not reset on *rdev* because it is not a shared DASD.
- HCP2001E The reserve was not reset on DASD *rdev* because it was not reserved.
- HCP2009I The system detected I/O errors while attempting to release a real reserve that was held on device *rdev* (by user *userid*, virtual device *vdev*). *{The reserve is now held by the system|The current reserved state of the real device could not be determined}*

# RESTART MSGPROC

► RESTART — MSGProc — *msgprocid* ◄

## Authorization

Privilege Class: A, B, G

## Purpose

The RESTART MSGPROC command allows a user ID to reload and start a message processor in a CF Service Machine.

If you have Class G privilege only, the message processor that you specify must be currently defined in your I/O configuration. A message processor may be defined in your I/O configuration by either the DEFINE MSGPROC command or the SPECIAL directory statement.

The RESTART MSGPROC command will cause the specified CF Service Machine to be loaded with the Coupling Facility Control Code (Licensed Internal Code) from the service element of the processor. Once the CF Service Machine is loaded, the message processor is restarted and all previous structures within the Coupling Facility will be lost.



**Attention:** This command affects all coupled guests connected to the message processor.

## Operands

### *msgprocid*

specifies the user ID of the CF Service Machine to be restarted.

## Usage Notes

1. The CF Service Machine's console may be viewed to verify that the restart completed successfully.
2. To define a virtual message processor and associated virtual message devices, use the DEFINE MSGPROC command.
3. To display information about virtual message processors in your I/O configuration, use the QUERY VIRTUAL MSGPROC command.
4. To display information about the virtual message devices within your I/O configuration, use the QUERY VIRTUAL MSGDEVICES command.
5. To remove all message facility capability from your virtual machine, use the SET MSGFACIL OFF command.
6. To detach a virtual message processor and associated virtual message devices, use the DETACH MSGPROC command.
7. If an IPL directory statement or the IPL operand on the XAUTOLOG command was initially used to load the CF Service Machine, then the same IPL statement will be used by the RESTART MSGPROC command when performing the restart. The CF Service Machine will not be loaded from the service element of the processor.

## Responses

Response 1:

## RESTART MSGPROC

To restart message processor CFCC1 within user SHARE1's I/O configuration, enter the following:

```
restart msgp cfcc1
HCP2809I Restart of message processor CFCC1 by SHARE1 initiated.
Ready;
```

HCP2809I will be displayed on both the message processor and the issuer's console.

### Messages

- HCP002E Invalid operand – *operand*
- HCP003E Invalid option – command contains extra option(s) starting with *option*
- HCP020E Userid missing or invalid
- HCP2805E Message processor *msgprocid* does not exist in your I/O configuration.
- HCP2809I Restart of message processor *msgprocid* by *userid* initiated.



## REWIND

```
▶▶ REWInd — vdev ◀◀
```

### Authorization

Privilege Class: G

### Purpose

Use REWIND to rewind, but not unload, tape on a real tape drive.

### Operands

#### *vdev*

is the virtual tape drive on which the tape is mounted.

### Usage Notes

If the tape drive specified is in a given state, the issuer must be the user ID to whom the tape drive was given, or the command cannot process.

### Responses

Response 1:

```
REWIND NOT PERFORMED
```

is returned when the rewind operation is pending.

Response 2:

```
REWIND COMPLETE
```

is returned when the rewind operation is complete.

### Messages

- HCP006E Invalid device type - *{rdev/vdev/ldev}*
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP263E Too many operands were supplied for this command.
- HCP1122E *command* was not performed. Tape *vdev* has been given to user *userid*.
- HCP1202E Device *vdev* was not rewound because it is active.

## SAVESEG

```
▶▶ SAVESEG — dcssname ▶▶
```

### Authorization

Privilege Class: E

### Purpose

SAVESEG is the final step of a saved segment build process. This command allows a system administrator or an installation procedure to save the page range areas previously specified by a DEFSEG command. The areas contain the appropriate objects, such as code or data, that were loaded into these defined areas by an install procedure.

Use SAVESEG to save the pages associated with the name of the saved segment into a previously defined class S (skeleton) system data file (SDF). Upon successful completion of SAVESEG, the class S SDF becomes an active class A (available) or class R (restricted available) SDF.

If the saved segment was defined as a member of a segment space, the segment space class S SDF has its member saved segment entry in the segment space's descriptor page updated to saved status. When all members of a segment space indicate saved status, the segment space class S SDF becomes an active class A or class R SDF.

The saved segment may have been defined as a member of several segment spaces. In this case, each segment space having this member would have its descriptor page updated.

### Operands

#### *dcssname*

is the name (a 1- to 8-character alphanumeric string) of the saved segment to be saved. This is the file name of a class S (skeleton) SDF previously defined with the DEFSEG command. When a DEFSEG command using the SPACE operand is issued, the *dcssname* specified in the SAVESEG command is interpreted as the member name specified in the DEFSEG command.

### Usage Notes

1. CP can process only one DEFSEG or SAVESEG command at a time. If CP is already processing a DEFSEG or SAVESEG command, processing of a subsequent DEFSEG or SAVESEG request is delayed. The SAVESEG command writes all page ranges, except those defined as EN or SN, to the associated system data file. The amount of time it takes for the command to complete is directly proportional to the amount of data to be written.
2. Using SAVESEG at off-peak hours on the system will have less impact on users due to I/O contention to the spool DASD.
3. You can interrupt and end a SAVESEG operation before it completes by using the PURGE NSS command. This will not leave a partially-saved system data file on the system. The skeleton system data file will be purged and must be redefined before the saved segment can be resaved.
4. After the execution of this command, the system data file is designated class A or class R. Class A indicates that it is in the unrestricted available state and may be accessed by any user. Class R indicates that the DCSS or segment space is restricted; to load it, the user must have a NAMESAVE statement for the saved segment in the directory entry.
5. Only one available (class A or R) saved segment with a particular name may exist at a given time. If any class A or R saved segment already exists with the same name as the one being saved, and there are no virtual machine users associated with the saved segment, the old copy of the saved segment

is purged, and the new one is made available when it becomes active (see usage note “11” on page 1395).

If users are associated with the old copy and neither the old copy nor the new copy of the saved segment is a CPDCSS (that is, no page ranges were defined with the SC page descriptor code), the old copy is marked as pending purge, and the new copy is saved. Any users associated with the old copy continue to reference the old (now pending purge) saved segment. Subsequent users are associated with the new copy.

If users are associated with the old copy and either the old copy or the new copy of the saved segment is a CPDCSS (that is, one or more page ranges were defined with the SC page descriptor code), the SAVESEG command is not accepted.

6. If there is a copy of the saved segment that is class P pending purge (the PURGE NSS command was used to purge it while there were users associated with it), the same rules apply as described in item “5” on page 1394:
  - If neither the old copy nor the new copy of the saved segment is a CPDCSS, the new copy *may* be saved
  - If either the old or the new copy is a CPDCSS, the new copy *cannot* be saved.
7. Saved segments are loaded through DIAGNOSE X'64'. You can also use the CMS SEGMENT command or macro, which call DIAGNOSE code X'64'. However, CMS does not support saved segments above 2047 MB. A DCSS that contains pages above 2047 MB can be loaded only through DIAGNOSE code X'64', using a 64-bit subcode.
8. A member saved segment cannot be loaded until at least one of the segment spaces of which it is a member is class A or R.
9. The contents of and storage keys for pages defined as or defaulted to "no data saved" on the DEFSEG command are not saved in the saved segment system data file when the SAVESEG command is processed. When referenced by a user who loads the saved segment, the storage contents and keys are zero. In this case, the page contents must be initialized after the DCSS is loaded.
10. When using the collaborative memory management assist, storage in the page ranges of a DCSS should be in the stable block-usage state at the time of the SAVESEG command. Otherwise, the SAVESEG operation might fail (with message HCP445E), or the contents and the block-usage and block-content states of that storage will be unpredictable when the storage is subsequently loaded into a virtual machine by DIAGNOSE X'64'.
11. The SAVESEG command may not be issued for a saved segment represented by *spacename* in a DEFSEG command (see “DEFSEG” on page 246).
12. The segment space SDF is changed from class S to class A when all members within the segment space have been saved.
13. If you use DEFINE STORAGE CONFIGURATION and define more than one storage extent, SAVESEG will save pages in page ranges that are located in the second and higher storage extents. However, if the pages are in a CP segment that has never been referenced (that is, no data was stored in an entire megabyte included within one of the page ranges defined on the DEFSEG command), the SAVESEG will fail.
 

In addition, you cannot load the saved segment after it is saved until the storage configuration is changed so that the saved segment loads below the start of the second defined storage extent.
14. Processing of a SAVESEG command might be delayed if DIAGNOSE code X'64' is concurrently performing the initial load of a large DCSS.

## Messages

- HCP263E Too many operands were supplied for this command.
- HCP300E Save request beyond virtual machine size
- HCP427E *type {rdev/vdev/name[user id]}* system spool space full; file {closed|held|purged}

## SAVESEG

- HCP435E SAVESEG command was not completed because a paging I/O error occurred while writing a page.
- HCP440I Saved segment *name* was successfully saved in fileid *fileid*.
- HCP445E The saved segment *name* could not be saved due to an unexpected error.
- HCP446E The saved segment *name* was not previously defined and cannot be saved.
- HCP448E System *name* not quiesced. Outstanding I/O on device *rdev*.
- HCP448E System *name* was not quiesced due to an outstanding external interrupt.
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*.
- HCP1359E [CP] Saved segment *name* cannot be saved because a [CP] copy is already {active| waiting to be purged}.
- HCP1364E Segment *name* is a segment space and will be saved by CP when the final member is saved.
- HCP1365E *name1 spoolid* has an entry for *name2* which was not found, [purge *name1*] purge *name1* or define *name3*|*name1* purged].
- HCP1374I The SAVESEG command was terminated by User *userid*. SEGMENT *name* was purged.

## SAVESYS

```
▶▶ SAVESYS — name ◀◀
```

### Authorization

Privilege Class: E

### Purpose

Use SAVESYS to save a named saved system (NSS) into a previously defined system data file (SDF).

### Operands

#### *name*

is the name of the NSS to be saved. This is the file name of a class S (skeleton) system data file previously defined with the DEFSYS command.

### Usage Notes

1. CP can process only one DEFSYS or SAVESYS command at a time. If CP is already processing a DEFSYS or SAVESYS command, processing of a subsequent DEFSYS or SAVESYS command is delayed. The SAVESYS command writes all page ranges, except those defined as EN or SN, to the associated system data file. The amount of time it takes for the command to complete is directly proportional to the amount of data to be written.
2. Using SAVESYS at off-peak hours on the system will have less impact on users due to I/O contention to the spool DASD.
3. You can interrupt and end a SAVESYS operation before it completes by using the PURGE NSS command. This will not leave a partially-saved system data file on the system. The skeleton system data file will be purged and must be redefined before the NSS can be resaved.
4. After the execution of this command, the system data file is designated class A or class R. Class A indicates that it is in the unrestricted available state, and may be accessed by any user. Class R indicates that the NSS is restricted and the users must have a NAMESAVE statement for the NSS in their directory entries in order to load it.
5. Only one available NSS (Class A or R) with a particular name may exist at a given time. If any Class A or R NSS already exists with the same name as the one being saved, and there are no virtual machine users associated with the NSS, then the old copy of the NSS is purged and the new one is made available.

If there are users associated with the old copy and neither the old copy nor the new copy of the NSS is a CP system service NSS (that is, no page ranges were defined with the SC page descriptor code), the old copy is marked as pending purge, and the new copy is saved. Any users associated with the old copy continue to reference the old (now pending purge) NSS. Subsequent users are associated with the new copy.

If there are users associated with the old copy and either the old copy or the new copy of the NSS **is** a CP system service NSS (that is, one or more page ranges were defined with the SC page descriptor code), the SAVESYS command is not accepted.

If there is a CP NSS with the specified name that is class P pending purge (the PURGE NSS command was used to purge it while there were users associated with it), the same rules apply as described in [“5” on page 1397](#):

- If neither the old copy nor the new copy is a system service NSS, the new copy *may* be saved

- If either the old or the new copy is a CP system service NSS, the new copy *cannot* be saved.
6. If a skeleton system data file exists for an NSS with the specified system name, the data is saved into it. If no skeleton NSS file exists, but there is a DCSS skeleton file with the same name, the data and storage keys (but not the virtual machine environment) are saved into the DCSS skeleton file, exactly as if the SAVESEG command had been issued.
  7. The contents of and storage keys for pages defined as or defaulted to "no data saved" on the DEFSYS command are not saved in the NSS system data file when the SAVESYS command is processed. When referenced by a user who IPLs the NSS, the storage contents and keys are zero. In this case, the page contents must be initialized after the NSS is IPLed.
  8. When using the collaborative memory management assist, storage in the page ranges of an NSS should be in the stable block-usage state at the time of the SAVESYS command. Otherwise, the SAVESYS operation might fail (with message HCP445E), or the contents and the block-usage and block-content states of that storage will be unpredictable when the storage is subsequently loaded into a virtual machine by IPL.
  9. An NSS is loaded through the IPL command.
  10. If the MACHMODE operand was used on the DEFSYS command that defined the NSS, then SAVESYS can be used to save the NSS only when the guest's MACHINE setting is one that was specified by the MACHMODE operand. If the NSS was defined without the MACHMODE operand, then SAVESYS can be used regardless of the guest's MACHINE setting.
  11. When an NSS is defined using a MACHMODE operand with more than one value, the NSS is known as a multimode NSS. The following architectural entities are saved by SAVESYS and restored by IPL of a multimode NSS:
    - Storage and storage keys
    - Program Status Word (PSW)
    - General purpose registers
    - Floating-point registers
    - Floating-point control register
    - Prefix register

Note that the saving and restoring of storage is under control of page-descriptor codes, such as EW and EN, specified on the DEFSYS command. Also, the restored general register contents may be replaced with parameter data specified by the PARM operand on the IPL command. See the description of the PARMREG operand of the DEFSYS command and the PARM operand of the IPL command for more information.

Immediately after the NSS is restored by the IPL command, all other architectural entities not mentioned above, such as timers or other registers, have the initial-CPU-reset values appropriate for the MACHINE setting of the guest in which the NSS was IPLed.

12. When an NSS is defined using a MACHMODE operand with a single value, or without the MACHMODE operand, the NSS is known as a single-mode NSS. The data saved and restored by SAVESYS is the same as that for a multimode NSS, but will also include the following:
  - Timers
  - Access registers
  - Control registers
13. The virtual machine's architecture mode is determined by its MACHINE setting. In some cases, the virtual machine's architecture mode is also determined by the architecture mode when the NSS was saved. The interaction of the MACHINE setting and the architecture mode when the NSS was saved has the following outcomes:

**MACHINE setting: ESA**

**Architecture mode when NSS was saved: ESA/390 or ESA/XC**

If the MACHINE setting is ESA and the NSS was saved while in ESA/390 mode or ESA/XC mode, then the architecture mode is ESA/390.

**Architecture mode when NSS was saved: z/Architecture or z/XC**

If the MACHINE setting is ESA and the NSS was saved while in z/Architecture mode or z/XC mode, then the architecture mode is z/Architecture.

**MACHINE setting: XC****Architecture mode when NSS was saved: ESA/390 or ESA/XC**

If the MACHINE setting is XC and the NSS was saved while in ESA/390 mode or ESA/XC mode, then the architecture mode is ESA/XC.

**Architecture mode when NSS was saved: z/Architecture or z/XC**

If the MACHINE setting is XC and the NSS was saved while in z/Architecture mode or z/XC mode, then the architecture mode is z/XC. However, if the virtual machine's relocation domain excludes z/XC capability, then IPL sets the architecture mode to ESA/XC.

**MACHINE setting: Z**

If the MACHINE setting is Z, then the architecture mode is z/Architecture.

The issuer of the SAVESYS command is responsible for ensuring that the saved program can operate correctly if it is resumed with the saved PSW in any of the architecture modes that are allowed by the MACHMODE option.

14. When CMS is built, the SAVESYS must be done in concert with an IPL. The IPL should be done as follows:

```
IPL cuu PARM SAVESYS
```

See *z/VM: Installation Guide* for information on how to SAVESYS the CMS nucleus.

15. If you use DEFINE STORAGE CONFIGURATION and define more than one storage extent, SAVESYS will save pages in page ranges that are located in the second and higher storage extents. However, if the pages are in a CP segment that has never been referenced (that is, no data was stored in an entire megabyte included within one of the page ranges defined on the DEFSYS command), the SAVESYS will fail.

In addition, you cannot load the NSS after it is saved until the storage configuration is changed so that the NSS loads below the start of the second defined storage extent.

16. Processing of a SAVESYS command might be delayed if DIAGNOSE code X'64' is concurrently performing the initial load of a large DCSS.
17. The SAVESYS command does not save vector registers.

**Messages**

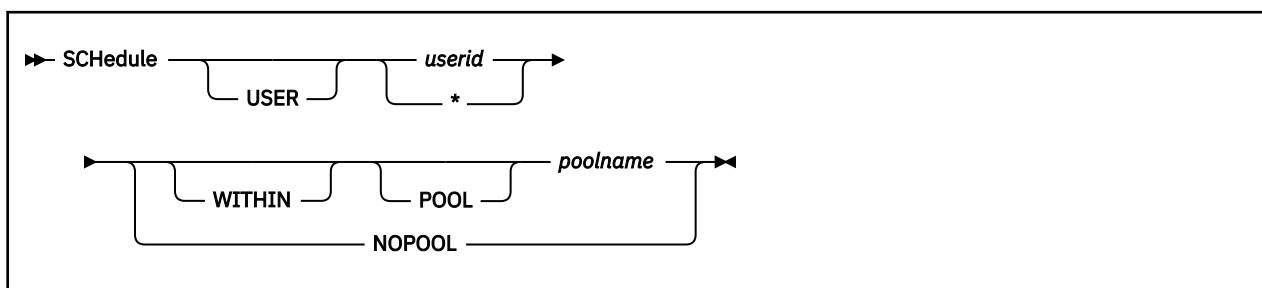
- HCP263E Too many operands were supplied for this command.
- HCP300E Save request beyond virtual machine size
- HCP427E *type* {*rdev*/*vdev*/*name*[*userid*]} system spool space full; file {closed|held|purged}
- HCP435E SAVESYS command was not completed because a paging I/O error occurred while writing a page.
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP440I Named Saved System (NSS) *name* was successfully saved in fileid *fileid*.
- HCP445E The Named Saved System (NSS) *name* could not be saved due to an unexpected error.
- HCP446E The Named Saved System (NSS) *name* was not previously defined and cannot be saved.
- HCP448E System *name* not quiesced. Outstanding I/O on device *rdev*.
- HCP448E System *name* was not quiesced due to an outstanding external interrupt.
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*.
- HCP1350E The saved segment *name* cannot be saved because MP users cannot save a Named Saved System (NSS).

## SAVESYS

- HCP1359E [CP] Named Saved System (NSS) *name* cannot be saved because a [CP] copy is already {active| waiting to be purged}.
- HCP1365E *name1 spoolid* has an entry for *name2* which was not found, [purge *name1*| purge *name1* or define *name3*|*name1* purged].
- HCP1368E Named Saved System (NSS) *name* can not be saved when running in {an ESA|an XC|a Z} virtual machine.
- HCP1374I The SAVESYS command was terminated by User *userid*. SYSTEM *name* was purged.



## SCHEDULE



### Authorization

Privilege Class: A

### Purpose

Use SCHEDULE to assign a user to an already defined resource pool or to remove a user from the resource pool to which the user is currently assigned.

### Operands

*userid*

\*

identifies the user assigned to or removed from a resource pool. If an asterisk (\*) is specified, the user is the user ID issuing the command. The specified user ID must be logged on or disconnected when the SCHEDULE command is issued.

*poolname*

**POOL** *poolname*

**WITHIN** *poolname*

**WITHIN POOL** *poolname*

specifies the resource pool to which the user is assigned. The resource pool must be already defined on the z/VM system and the type of CPU being limited in that resource pool must match the user's primary CPU type. If the specified user is already assigned to another resource pool, the user is removed from the current resource pool and moved to the new resource pool. The CPU resource the user consumed prior to the move is charged against the old pool; future consumption will be charged against the new resource pool, and the user will be subject to the resource limits defined that pool. If the user is unable to move to the new resource pool for any reason, the user remains in the current pool. For information on defining resource pools, see [“DEFINE RESPOOL” on page 205](#)).

**NOPOOL**

specifies that the user is to be removed from whatever resource pool the user is in. After the user is removed from the resource pool, the user will no longer be subjected to pool limits. The remaining users in the resource pool can consume more resources, as there are fewer members in the limited group while the resource pool limit applied to them remains constant.

### Usage Notes

1. When adding a user to a resource pool via the SCHEDULE command, the user's primary virtual CPU type must match the group's virtual CPU type, and CPU affinity must be on for the user. Based on whether there is an IFL processor online for that system, z/VM will run this user ID (and the resource pool to which the user has been added) with CPU affinity set on or suppressed. You are allowed to add an all-IFL user to an IFL virtual CPU type group when there is no IFL processor on that system.
2. After a user is assigned to a resource pool, you cannot issue the SET CPUAFFINITY OFF command for that user, and CPU affinity is suppressed only when that user is running with a virtual IFL on a system where there are no IFL processors online.

## SCHEDULE

3. After a user is assigned to a resource pool, you cannot issue a DEFINE CPU command that would change the user's primary virtual CPU, because the user's primary virtual CPU type must continue to match the group's virtual CPU type.
4. Only a logged on or disconnected user can be assigned to a resource pool. Issuing the SCHEDULE command for a logged on or disconnected user assigns the user to the resource pool immediately, and the assignment remains in effect until another SCHEDULE command is issued for that user, or the user logs off. This allows an administrator to assign a user to a resource pool temporarily.

If a permanent assignment of a user to a resource pool is desired, the SCHEDULE command for the assignment can be placed in the user's directory entry using the COMMAND directory statement. If no other SCHEDULE command is issued, the user will remain in that resource pool until the user logs off. The SCHEDULE command can be placed anywhere where the COMMAND directory statement is allowed, including the directory profile for pooled users, where \* should be specified for the user ID to put the logging on user into the resource pool.

To have an assignment take effect immediately and be permanent, both of these steps must be done.

5. When setting up permanent resource pools and the assignment of users, the order of the commands is very important. To save CPU time at logon, it is recommended that only one SCHEDULE command be included in a user's directory entry. If the directory entry contains multiple SCHEDULE commands, they will all be processed in the listed order, with the user assigned to the resource pool specified on the last SCHEDULE command. The DEFINE RESPOOL command must be issued to create the resource pool before the SCHEDULE command is issued. Any DEFINE CPU or SET CPUAFFINITY commands in the user's directory entry must be executed before the SCHEDULE command in the directory entry, because they are restricted after the SCHEDULE command is executed. Because the defaults are SET CPUAFFINITY ON and SET SRM LIMITHARD CONSUMPTION, resource pooling does not require a lot of setup commands to be issued via the directory.
6. Any individual CPU limits on the user as defined by the SET SHARE command remain in effect. If a user with an individual CPU limit is assigned to a resource pool, both limits will be checked each time the user's CPU use is evaluated, and the more restrictive limit will be applied.
7. Share settings control access to CPU power on a system-wide, per-processor-type basis, but they have no bearing or influence on how the power associated with a RESPOOL is distributed to its members. When the RESPOOL limit is reached, all members of the pool are put onto the limit list for a while, and then, after a delay sufficient to compensate for any excess use, they are all removed. The scheduler makes no effort to distribute the power of the RESPOOL to its members in accordance with the members' share settings.
8. A user assigned to a resource pool can be relocated only to another member of an SSI cluster that contains a resource pool with the same name and the same type of virtual CPU being limited. All users in the resource pool need to use the same underlying CPU resource for resource pooling to work, which is why all users assigned to a resource pool must have CPU affinity on and can relocate only to a resource pool with the same virtual CPU type as the current resource pool. Based on the availability of IFL CPUs on the destination system, z/VM will toggle the relocated user between CPU affinity on and CPU affinity suppressed, as necessary. Because the scope of a resource pool is a single z/VM system, the relocation will essentially remove the user from the resource pool on the source system and assign the user to the resource pool on the destination system. When a VMRELOCATE TEST or MOVE command is issued against a user ID, the SCHEDULE command cannot be issued to move the user to a new resource pool until the VMRELOCATE command processing has ended.

## Responses

Response 1:

```
User userid has been added to resource pool poolname
```

confirms that the user who was not in a resource pool has been added to the pool as requested.

Response 2:

```
User userid has been removed from resource pool poolname
```

confirms that the user has been removed from the resource pool as requested and will now be scheduled as an individual user only.

Response 3:

```
User userid has been removed from resource pool poolname
User userid has been added to resource pool poolname
```

confirms that the user who was already in a resource pool has been moved to a new resource pool as requested. Two responses are issued to indicate that the user was removed from one resource pool and added to another resource pool. However, this action occurs as a single step, and there is no point in time when the user is not in any resource pool.

Response 4:

```
User userid is already in resource pool poolname
```

indicates that the user is assigned to the resource pool as requested. The user was not added to or removed from any resource pool because the user was already in a resource pool before the SCHEDULE command was issued.

Response 5:

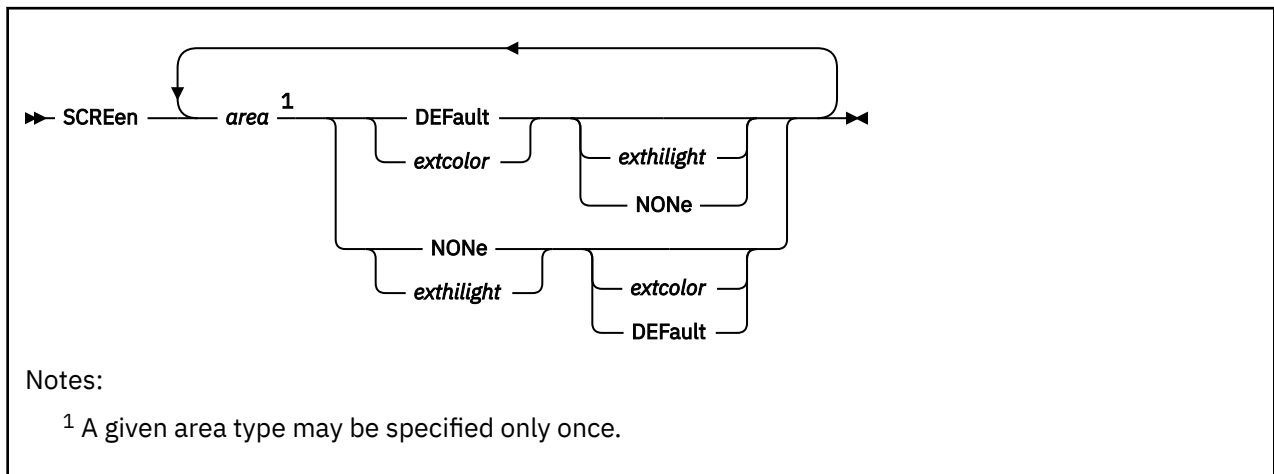
```
User userid is not in a resource pool
```

indicates that the user is not assigned to the resource pool as requested and was not in any resource pool when the SCHEDULE command was issued.

## Messages

- HCP002E Invalid operand — *operand*
- HCP003E Invalid option — command contains extra option(s) starting with *option*
- HCP007E Invalid userid — *userid*
- HCP045E *userid* not logged on
- HCP1001E An operand is missing for USER
- HCP1001E An operand is missing for POOL
- HCP3151E Resource pool *poolname* does not exist
- HCP3153E User *userid* is not eligible for resource pool *poolname*
- HCP3156E Resource pool assignment cannot be changed for user *userid* while relocating
- HCP3157E User *userid* must have CPU Affinity ON to be put into a resource pool
- HCP6704E Missing token at end of line

## SCREEN



### Authorization

Privilege Class: G

### Purpose

Use SCREEN to change:

- Screen color definition
- Screen highlight definition.

The initial settings are the default colors green and white.

### Operands

#### *area*

specifies the area of the screen. The area can be:

<b>Area</b>	<b>Meaning</b>
ALL	The entire screen. If you specify this operand, you cannot specify any other area on the same command.
INArea	The input area
STAtarea	The system status area
OUTarea	The output area. If you specify OUTAREA, you cannot specify CPOUT, VMOUT, or INREDISP on the same command line.
CPOut	Output from the CP. You may not specify CPOUT on the same command as OUTAREA.
VMOut	Output from CMS or the virtual machine operating system running in your virtual machine. You may not specify VMOUT on the same command line as OUTAREA.
INRedisp	Your input redisplay. You may not specify INREDISP on the same command line as OUTAREA.

#### *extcolor*

#### **DEFAult**

specifies the color you want to assign to an area of your screen. The choice of colors are:

- BLUe
- TURquois
- RED
- PINK
- GREen
- WHIte
- YELlow
- DEFault (green and white)

***exthilight*****NONE**

specifies the extended highlight value you want to assign to an area of your screen. The *exthilight* value can be:

<b><u>Value</u></b>	<b><u>Meaning</u></b>
BLInk	Blinking
REVvideo	Reverse video
UNDerlin	Underlining
NONE	No extended highlighting

**Usage Notes**

1. If you enter the SCREEN command from a display that does not support extended color or extended highlighting, CP still sets the values you specify for the virtual machine console. If you disconnect from that display and logon at a display that does support extended color or extended highlighting, the values you set are in effect for your virtual machine after the reconnection. These values remain in effect for the duration of the current session unless you alter them with another SCREEN command.
2. You cannot specify a particular *area* operand more than once on the same SCREEN command line.
3. Since the command is not device dependent, you may enter the SCREEN command from any IBM-supported terminal or from a PROFILE EXEC. However, the terminal controller must have the extended color feature installed for the SCREEN command to have an effect on your display.
4. If you enter the SCREEN command for a particular area and you specify only the *extcolor* or default option, the *exthilight* remains unchanged from its previous setting. Likewise, if only the *exthilight* or NONE option is specified, the *extcolor* remains unchanged from its previous setting.
5. If CPOUT uses the default color, then incoming MSGs, WNGs, and MSGNOHs will be displayed in white. If you change the CPOUT color, then incoming MSGs, WNGs, and MSGNOHs are displayed in the CPOUT color you specified.

**Examples**

If you enter

```
screen outarea red none inarea blink green statarea pink underlin
```

CP sets the following values for your terminal:

<b>Area specified</b>	<b>Area affected</b>	<b>Color</b>	<b>Exthilight</b>
outarea	cpout	red	none
	vmout	red	none
	inredisp	red	none
inarea	inarea	green	blinking

## SCREEN

Area specified	Area affected	Color	Exthighlight
statarea	statarea	pink	underlining

### Responses

Response 1:

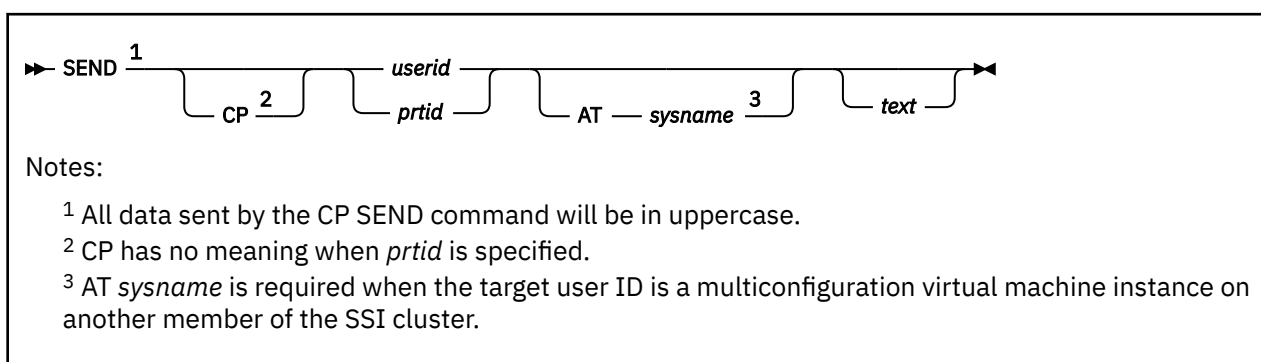
```
Command complete
```

is issued when the command completes successfully.

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid

## SEND



### Authorization

Privilege Class: C, G

### Purpose

Use SEND to send console input to a virtual machine. As a Class G user, you can send input to a disconnected virtual machine for which you provide secondary console services. As a Class C user, you can send input to any virtual machine. You may enter the following:

- CP commands
- Virtual machine commands
- Responses to virtual machine or CP messages
- Other virtual machine data, as required.

### Operands

#### CP

indicates that CP should process the input text on behalf of the receiving virtual machine. If the first four characters of the text you want to send are your logical escape character, the receiver's logical line end character, and CP, then you need not specify the CP operand to send a CP command. The same holds true if the receiver's virtual machine is already in CP READ. If you specify the CP operand with no text, the receiving virtual machine is placed in console function mode, just as if a #CP command were entered.

#### *userid*

identifies the virtual machine that is to receive the console input.

#### *prtId*

indicates that the command is to be passed for processing to the print server virtual machine designated by *prtId*. The print server virtual machine that receives and processes the SEND command also provides *prtId*. The operand *prtId* is one to eight alphanumeric characters. If you specify four or fewer characters for the *prtId*, *prtId* cannot contain all hexadecimal characters, because CP then interprets *prtId* as *rdev*.

#### AT

indicates that the target user ID is on another member of the SSI cluster. AT *sysname* is required if the target user ID is a multiconfiguration virtual machine instance on another member of the SSI cluster.

#### *sysname*

is the system name of an active member in the SSI cluster.

## SEND

### *text*

is the data that is passed to the disconnected virtual machine as console input. Mixed DBCS data may be used as input text.

## Usage Notes

1. Use a **send cp userid** command (without *text*) to force a CP READ when no CP READ exists.
2. If you are a secondary user, you might receive one of the following messages from the primary user:

```
HCP150A User userid has issued a CP read
HCP150A User userid has issued a VM read
HCP150A User userid has issued a CP or VM read.
```

To reply to the message, issue one of the following SEND commands:

```
send userid text
send cp userid text
```

Whether you need to specify the CP operand on the SEND command depends on the following principles:

- If there is no CP read or VM read pending on the primary user's console, then SEND (without CP) will be posted as unsolicited input to either CP or the virtual machine, based on the TERM MODE setting. This is analogous to typing input when the screen shows RUNNING status.
- If there is a CP read pending on the primary user's console, then SEND (without CP) will satisfy that CP read. This is analogous to typing input when the screen shows CP READ status.
- If there is a VM read pending but no CP read pending on the primary user's console, then SEND (without CP) will satisfy that VM read. (A CP read supersedes a VM read.) This is analogous to typing input when the screen shows VM READ status.
- SEND CP always either satisfies a CP read, if one is pending, or is presented as an unsolicited CP command. This is analogous to typing input prefixed with #CP.

The different cases are shown in [Table 33 on page 1408](#).

<b>CP Read Pending</b>	<b>VM Read Pending</b>	<b>TERM MODE</b>	<b>Effect of SEND <i>userid text</i></b>	<b>Effect of SEND CP <i>userid text</i></b>
No	No	VM	input goes to virtual machine	input goes to CP
No	No	CP	input goes to CP	input goes to CP
No	Yes	VM	input satisfies VM read	input goes to CP
No	Yes	CP	input satisfies VM read	input goes to CP
Yes	No	VM	input satisfies CP read	input satisfies CP read
Yes	No	CP	input satisfies CP read	input satisfies CP read
Yes	Yes	VM	input satisfies CP read	input satisfies CP read
Yes	Yes	CP	input satisfies CP read	input satisfies CP read

**Notes:**

- a. "input goes to CP" means the input is treated as unsolicited input and therefore is treated the same as a #CP command typed at the console.
- b. "input goes to virtual machine" means the input is treated as unsolicited input; it causes an attention interruption on the virtual console, and is supplied as the response to the line mode read which the guest is expected to post in response to that interruption.



3. If you are a Class G user, then:
  - a. The receiving virtual machine must be disconnected at the time you enter the SEND command.
  - b. You must be authorized as the secondary user on the CONSOLE statement in the directory entry of the receiving virtual machine or authorized as a result of the receiving virtual machine's use of the CP SET SECUSER command.
  - c. The secondary user must not be disconnected when the SEND command is issued, unless the secondary user has a valid path to the \*MSG or \*MSGALL system service.
4. If you are not the receiving virtual machine's secondary user, you do not receive responses to the commands you send it.
5. If both a CP read and a VM read are pending, the CP read is satisfied first, regardless of the order in which they were posted.
6. If you want to send multiple lines of input on a single SEND command, use the receiver's logical-line-end character preceded by your logical-escape character to separate the lines. The receiving virtual machine must also have line editing enabled. This restriction applies to when you SEND a #CP command as well.

If any of your logical-line-editing characters are part of the text to be sent to the disconnected virtual machine, they must be preceded by your logical-escape character.

7. For *prtId*, the virtual machine associated with the *prtId* processes the specified text. Therefore, the text on the SEND command must be valid for the virtual machine that is doing the processing.
8. You cannot use the class C version of this command if an external security manager is installed on your system and security label checking is enabled. For additional information, contact your security administrator.
9. Within a single system, the FOR command is an alternative to the SEND command. The FOR command is used to issue CP commands to other virtual machines and provides responses and return codes to the issuer. The responses can be directed to the console or handled by an IUCV connection to the Asynchronous CP Command Response (\*ASYNCMD) system service. The target virtual machine does not see the command or command responses. See the description of [“FOR” on page 501](#) for further details.
10. If you are using the SEND command to send an AT command, you can preface the AT command with CP to avoid confusion with the SEND command AT operand.
11. In an SSI cluster, the secondary user and primary user can be logged on to different member systems.

If either user is a multiconfiguration virtual machine:

- The secondary user will function in that capacity only when it is local (logged on to the same member as the primary user). If the primary user is a multiconfiguration virtual machine, the secondary user must be logged on to the same member as the instance of the primary user for which the secondary user setting was established:
  - If the setting was established by a CONSOLE statement in the identity entry in the primary user's virtual machine definition, a potential secondary user relationship exists for every member instance of the primary user.
  - If the setting was established by a CONSOLE statement in a subconfiguration entry in the primary user's virtual machine definition, a potential secondary user relationship exists for that member instance of the primary user.
  - If the setting was established by a SET SECUSER command, a potential secondary user relationship exists for the member on which the SET SECUSER command was issued.
- If the secondary user is remote:
  - The secondary user can issue SEND commands to the primary user. The AT *sysname* operands are required if the primary user is a multiconfiguration virtual machine instance.
  - The secondary user will not receive responses to SEND commands or any other output from the primary user.

## SEND

### Examples

1. Enter **send cp user1** to place the disconnected user, USER1, into console function read.
2. Enter **send user3 msg user2 hi** to cause USER2 to receive the message *hh:mm:ss: \*MSG FROM USER3 HI*.
3. If the secondary user receives the message *USER5 CHANGE TOD CLOCK (YES|NO)* and responds by entering **send user5 yes**, the receiving virtual machine accepts this input as if it were entered from the receiver's own terminal.
4. To send information to a specific printer virtual machine, enter **send prtId query prtname options**.

#### **prtId**

identifies the printer.

#### **QUERY prtname options**

identifies the text.

### Messages

- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP068E SEND command failed; *text*
- HCP1149E Error communicating with another SSI member. It cannot be determined if *command* executed remotely.
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.

# SET

---

## Purpose

Use the SET command to control various functions associated with your virtual machine or with the z/VM system.

## Usage Notes

1. Use the QUERY SET command to determine the current settings of all the following functions:

ACNT*	NOPDATA
AFFINITY*	NOTRAN*
ASSIST*	PAGEX
AUTOPOLL	RUN
CCWTRAN*	SMSG
CONCEAL	STBYPASS*
CPCONIO	STMULTI*
DIALDROP	SVCACCL*
ECMODE*	SVC76
EMSG	TIMEBOMB
IMSG	TIMER
ISAM*	VMCONIO
LINEDIT	VMSAVE*
MACHINE	WNG
MIH	370ACCOM
MSG	370E*

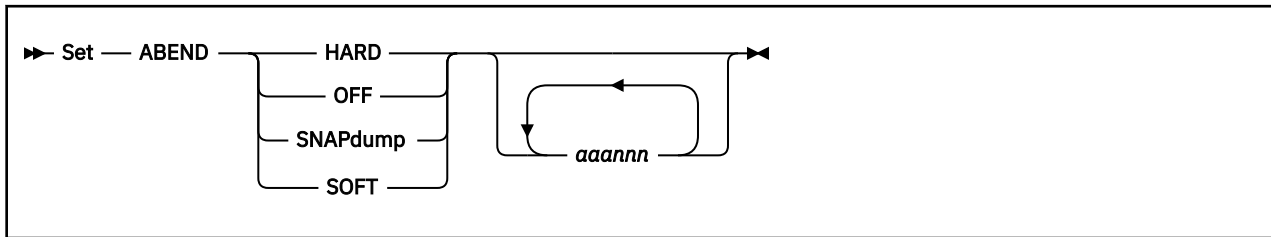
\* For compatibility purposes only – you cannot use the SET command to change these values, as they are either fixed or no longer valid.

2. SET ECMODE ON is supported for compatibility only. If any option other than ON is specified, the message 'INVALID OPTION' is issued.
3. Use the QUERY command with the appropriate operand to determine the setting of each of the following functions:

## SET

ABEND	
ACCOUNT	
AGELIST	
CACHE	
CACHEFW	
CFLINKS	
CMDLIMIT	
CONFIGMODE	
CPCHECKING	
CPLANGUAGE	
CPPROTECT	
CPTRACE	
CPUAFFINITY	
CPUID	
CRYPTO	
CU	
D8ONECMD	
DASD	
DASDFW	
DATEFORMAT	
DUMP	
DUMPDEV	
DYNAMIC_IO	
EDEVICE	
ENCRYPT	
HOTIO	
IOASSIST	
IOCDS_ACTIVE	
IO_OPT	
IOPRIORITY	
IPLPARMS	
JOURNAL	
KEYALIAS	
LAN	
LKFAC	
LKFACR	
LOADDEV	
LOGMSG	
LPARS	
LSYSTEM	
MAXLDEV	
MAXUSERS	
MDCACHE	
MEMASSIST	
MITIME	
MONDATA	
MULTITHREAD	
	NEW_DEVICES
	NIC
	NVS
	OBSERVER
	PAGING
	PASSWORD
	PF
	PRIVCLASS
	PRODUCT
	PROMPT
	PSWTRANS
	QIOASSIST
	QUICKDSP
	REORDER
	RESERVED
	RESPOOL
	RETRIEVE
	SCMEASURE
	SECUSER
	SHARE
	SHUTDOWNTIME
	SIGNAL
	SIGNALS
	SRM (DSPBUF)
	SRM (DSPSLICE)
	SRM (IABIAS)
	SRM (LDUBUF)
	SRM (MAXWSS)
	SRM (STORBUF)
	SSI
	STGEXEMPT
	STGLIMIT
	STORAGE
	SYSCONTROL
	SYSOPER
	THROTTLE
	TIMEZONE
	TOKEN
	TRACEFRAMES
	UNDERScore
	VARIABLE
	VDISK
	VMLAN
	VSWITCH
	VTOD
	WRKALLEG

## SET ABEND



### Authorization

Privilege Class: B

### Purpose

Use SET ABEND to force soft errors to cause system termination with hard abend format dump, full system snapdumps, or to prevent soft abend dumps from being generated.

### Operands

#### HARD

specifies that any soft abend causes full system termination with a dump and then attempts to restart the system.

#### HARD *aaannn*

specifies the soft abend with abend code *aaannn*, where *aaa* is the three character module identifier and *nnn* is the three character abend code number, causes full system termination, but other soft abends continue to be processed normally.

#### OFF

specifies that any soft abend will not generate a soft abend dump. A symptom record will be created to provide a record of the abend.

#### OFF *aaannn*

specifies the soft abend with abend code *aaannn* will not generate a soft abend dump. A symptom record will be created to provide a record of the abend.

#### SNAPDUMP

specifies that any soft abend causes a full system snapdump.

#### SNAPDUMP *aaannn*

specifies the soft abend with abend code *aaannn* causes a full system snapdump, but other soft abends continue to be processed normally.

#### SOFT

cancels all previous specifications of HARD, HARD *aaannn*, OFF, OFF *aaannn*, SNAPDUMP and SNAPDUMP *aaannn*. All soft abends operate normally as established during system initialization.

#### SOFT *aaannn*

specifies the soft abend with abend code *aaannn* will generate a soft abend. This effectively cancels any previous specification of HARD *aaannn*, OFF *aaannn*, or SNAPDUMP *aaannn*. Also, SOFT *aaannn* will override any global HARD, OFF or SNAPDUMP specification for the soft abend with abend code *aaannn*.

### Usage Notes

1. If any invalid abend code (operand does not have *aaannn* format) is encountered when a specific abend code is specified with any SET ABEND operand, no soft abend settings are changed.
2. You can specify up to 20 unique soft abend codes with one or more invocations of the SET ABEND command.

## SET ABEND

3. The abend codes (*aaannn*) are inspected for format validity. The *aaa* piece consists of three characters that can be any combination uppercase letters A-Z, lowercase letters a-z, or numbers 1-9. The *nnn* piece is a three digit decimal number from 000 to 255. The abend code is not inspected to determine whether it corresponds to a soft abend that is actually present in the system.

Code that is running as part of CP requests abends by using the HCPABEND macro. If the part containing the HCPABEND macro does not start with 'HCP' then the name of the module is stored in lowercase in the HCPABEND macro expansion. When a dump is created, the first three characters of the abend code is in lowercase. This is done because it allows support personnel to quickly identify abends that are not generated from a standard IBM CP nucleus part. This lowercase letter convention affects code that is added to CP via the CPXLOAD command or by using an alternate MDLAT macro and the VMSES/E command GENCPBLS. By using upper or lowercase letters with abend codes specified on SET ABEND, HCP and non-HCP abends can be handled differently.

An abend code containing lowercase characters can be specified on the SET ABEND command that is passed to CP using the DIAGNOSE X'08' programming interface. This could be done in a REXX EXEC or an assembler program. Also, a lowercase abend code can be passed using "Address Command" in a REXX EXEC. However, the simplest way is to use CMS Pipelines. A pipe command like the following could be passed to CMS to invoke a CP command.

```
PIPE CP SET ABEND HARD bcd001 efg002 | console
```

Note that the "SET ABEND HARD" part of the command must be specified in uppercase so that it passes tests in the CP parsing code. Specify abend code(s) in upper or lowercase, as necessary.

4. There may be some unique circumstances which will prevent a dump from being created despite having the ABEND set to SNAPDUMP. This will only occur if SNAPDUMP cannot get the resource which guarantees that no other processor in the configuration will force the abending processor offline. If a dump was to be taken at this time, its contents would not accurately reflect the abending environment.
5. When SET ABEND is issued for HARD, OFF, or SNAPDUMP, the soft abend dump spool file is automatically closed. This is a class D spool file that is normally opened during system initialization and is owned by the user ID specified with the DUMP operand on the SYSTEM\_USERIDS configuration statement. It has a file type of "HCPDUMP". Once closed, it can be reopened by issuing SET ABEND SOFT or SET ABEND SOFT *aaannn*.

### Examples

Specific invocations are cumulative. For example:

```
SET ABEND HARD ABC001 BCD002
```

Will add soft abends ABC001 and BCD002 to the abend table, indicating that they should cause a hard abend if hit.

```
SET ABEND SNAP BCD002 CDE003 DEF004
```

The abend table will now have 1 soft abend set to hard (ABC001), and 3 soft abends set to snapdump (BCD002, CDE003 and DEF004).

```
SET ABEND SOFT CDE003
```

Will reset soft abend CDE003 from generating a snapdump abend back to a soft abend. The other abends are not affected.

Global invocations (no specific abend specified) reset all previously set specific abends. For example (assuming the preceding invocations are still in effect), suppose the following command is entered:

```
SET ABEND OFF
```

The preceding soft abends that have been set to hard, soft, and snapdump, along with all of the remaining soft abends, will not generate a dump.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP400I All {paging|spooling} space is in use.
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP847E Maximum system spool file limit exceeded
- HCP1011I Maximum number of unique inhibited soft abends exceeded.
- HCP9308I Spooling I/O error occurred while reading or writing a system data file.
- HCP9309E CP was unable to allocate a soft abend dump file.

## SET ACCOUNT

---

```
► Set — ACCount — nnnnnnnn ◄
```

### Authorization

Privilege Class: G

### Purpose

Use SET ACCOUNT to change the current session account number.

### Operands

**nnnnnnnn**

is the 1- to 8-character account number that replaces the current session account number.

### Usage Notes

1. The account number you specify on the command line must be the same as one contained in your entry in the system directory.
2. If the account number you specify is valid, an accounting record is created for your session under the current accounting number.

### Messages

- HCP026E Operand missing or invalid
- HCP266E An account number was not supplied or is invalid.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]



## SET ADJUNCTS



### Authorization

Privilege Class: A

### Purpose

The adjunct support for a system can be enabled or disabled with this command. It allows dynamic alteration of the setting established at system initialization or by the FEATURE ENABLE ADJUNCTS or FEATURE DISABLE ADJUNCTS statement in the system configuration file. If no such FEATURE statement exists, the support is enabled by default.

### Operands

#### ON

Enable adjunct support.

#### OFF

Disable the ability for users to create adjunct configurations via the ADJUNCT START command. If an adjunct configuration already exists when this command is issued, it can continue to operate; all ADJUNCT command options except START will be accepted until the adjunct is ended.

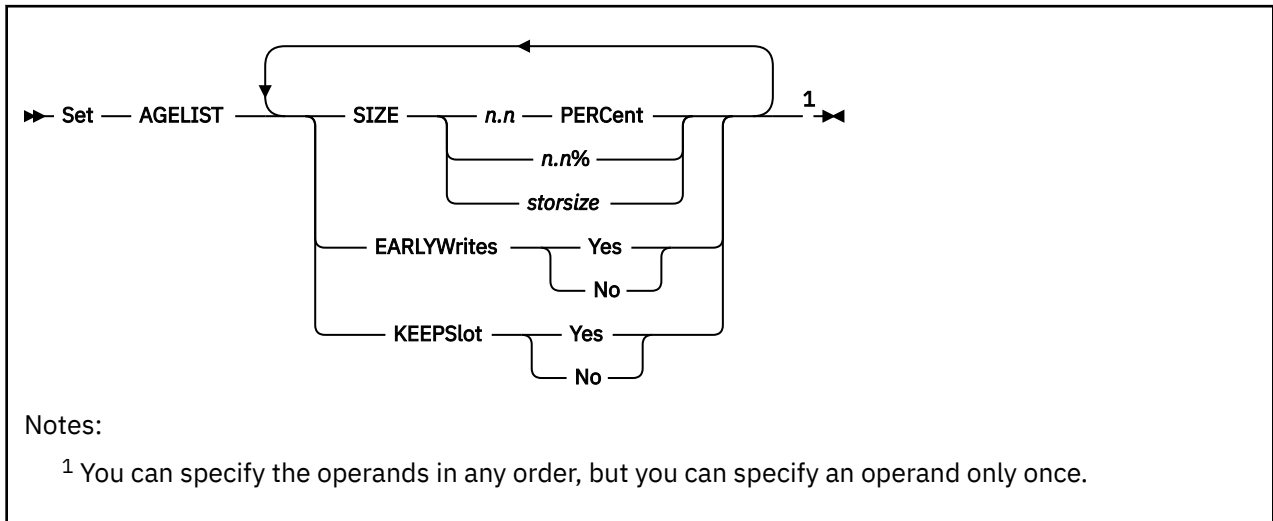
### Usage Notes

1. See the ADJUNCT directory statement in *z/VM: CP Planning and Administration* for more information on configuring an adjunct.
2. For a user to create an adjunct, ADJUNCTS must be set ON, and an ADJUNCT statement must exist in the user's directory entry (for example, ADJUNCT CMSADJ).

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid

## SET AGELIST



### Authorization

Privilege Class: A

### Purpose

Use SET AGELIST to modify the target size of the global aging list used by the system's frame replenishment algorithm and to tailor the algorithm's behavior in managing guest pages on auxiliary storage.

To satisfy the system's need for real storage, the frame replenishment algorithm selects candidate frames from the user and system frame-owned lists. The pages associated with the selected frames are put in the invalid-but-resident (IBR) state. Any access to a page in the IBR state by the guest results in a re-validation. Otherwise, after a period of aging on the local frame-owned list, the frame associated with an IBR page is added to the global aging list.

If a page is referenced while its frame is on the aging list, the page is re-validated and the frame moves back to the appropriate frame-owned list. Before a frame can be reclaimed from the aging list, the page content must be backed in auxiliary storage.

The target size of the aging list is a guideline to CP. The actual amount of storage in the global aging list at any point in time may differ significantly below the target amount, and at times may rise above the value.

### Operands

#### SIZE

is the target size of the global aging list used by the frame replenishment algorithm, specified as an amount of storage in one of the following formats.

#### *n.n* PERCent

#### *n.n%*

is the target size of the global aging list used by the frame replenishment algorithm, specified as a percentage of the current size of the dynamic paging area (DPA). The size of the DPA varies over time and it includes available frames and those that contain pageable content. When a percent value is specified, the system recalculates the target storage size periodically to adjust to the changing size of the DPA. The computed value is rounded up to the next megabyte boundary.

The decimal percentage can have at most one decimal place, and be in the range of .1 through the maximum allowed percentage of 5%. Regardless of the size of the configuration, CP never sets a target size smaller than 1M.

### **storsize**

The format of the storage size is *nu* where *n* is a 1-7 digit decimal number greater than 0 and *u* is the 1-character storage unit suffix (M,G,T,P,E). The specified storage size may not exceed 5% of the current size of the dynamic paging area.

### **EARLYWrites**

when Yes, the frame replenishment algorithm backs up page content to auxiliary storage in advance of frame reclaim to maintain a pool of readily reclaimable frames. When No is specified, pages are backed up only when the system is in need of frames.

### **KEEPSlot**

when Yes is specified, the auxiliary storage address (ASA) to which a page is written during frame replenishment will remain allocated to that page when it is read into real memory upon access by the guest. Preserving a copy of a page on auxiliary storage eliminates the need to rewrite the contents if the page is unchanged prior to the next steal operation. However, it could result in more fragmentation on the device over time.

When No is specified, the ASA for the page is unallocated during page fault processing. Releasing a paging ASA makes it available for reallocation and reduces the size of the auxiliary storage footprint.

## **Usage Notes**

1. It is suggested that the STORAGE AGELIST statement in the system configuration file be used to set the size of the aging list and the behavior of the frame replenishment algorithm to make sure the values are in effect during IPL. Any changes made to these settings after IPL using the SET AGELIST command will occur gradually. See the [STORAGE Statement in z/VM: CP Planning and Administration](#), for more information about the configuration statement and applicable defaults.
2. It is preferable to specify the target size of the aging list as a percentage of the DPA rather than a fixed amount. This allows CP to periodically adjust the size in response to system behavior.
3. If you increase the amount of real storage available to z/VM using the SET STORAGE command, you may also want to consider adjusting the target size of the aging list, especially when you have defined the size as a fixed amount.
4. Because specification of the aging list target size as a percentage of the DPA is rounded up to the next megabyte boundary, the resulting calculation may exceed the maximum value allowed as a fixed-size specification.
5. See QUERY AGELIST in *z/VM: CP Commands and Utilities Reference* for guidelines to determine an appropriate aging list size for your storage configuration and workload.
6. When the SET AGELIST command is used after IPL to change KEEPSlot from Yes to No, the auxiliary storage locations associated with any pages that are currently resident are not released. The new setting applies only to future reads from auxiliary storage. For this reason, it is recommended that you use the STORAGE AGELIST statement in the system configuration file to set KEEPSlot to No, rather than doing this after IPL using the SET AGELIST command.

To further eliminate copies of resident pages on auxiliary storage, you might want to also set EARLYWrites to No. However, doing so could reduce performance on a storage-constrained system because frame replenishment will not have a readily-available pool of frames to reclaim.

7. When a SET STORAGE command is issued to decrease the size of the storage configuration, a fixed AGELIST size will be reduced, if necessary, to equal the maximum allowed size of 5% of the dynamic paging area (DPA) size. If an adjustment is made, message HCP2597I is issued to the system operator. There is no automatic increase in the fixed AGELIST size when a SET STORAGE command is used to add storage to the configuration. If you want the AGELIST size to vary based on the size of the storage configuration, set the AGELIST size to be a percentage of the DPA size.

## Responses

```

Target size      = nU (mU) x.x% of pageable storage
In use          = nU
Pending writes  = nU
Early writes    = Yes|No
Sizing         = Fixed|Variable
Keep slot      = Yes|No

```

where:

### Target size *nU*

identifies the maximum expected amount of pageable storage to be contained in the global aging list. The target size is displayed in a storage unit (K,M,G,T,P,E) suitable to represent the values displayed in the remaining lines of the command response.

### (*mU*)

the target size in a storage unit with a precision of no more than 7 significant digits. This value may be rounded if the aging list size is defined as a percentage of pageable storage. When the aging list is empty this value is omitted.

### *x.x%* of pageable storage

identifies the target size as a percentage of pageable storage. The value may be rounded to the nearest tenth of a percent if the aging list size is defined as a fixed storage size.

### In use *nU*

identifies the amount of storage currently within the global aging list in storage units (K,M,G,T,P,E). Usage of the aging list is at the discretion of CP based on storage constraints. The amount of storage currently within the aging list may be significantly less than the target size, or in some cases slightly higher.

### Pending writes *nU*

identifies the portion of the global aging list in storage units (K,M,G,T,P,E), that must be backed up in auxiliary before the associated frames can be reclaimed.

### Early writes = Yes

indicates the frame replenishment algorithm backs up page content to auxiliary storage in advance of frame reclaim to maintain a pool of immediately reclaimable frames.

### Early writes = No

indicates the frame replenishment algorithm backs up page content to auxiliary storage only when the system is in need of frames.

### Sizing = Fixed

indicates the size of the aging list was specified on the SET AGELIST command as a fixed storage size.

### Sizing = Variable

indicates the size of the aging list was specified on the SET AGELIST command as a percentage of pageable storage. Since the number of real frames containing pageable content varies over time, so will the size of the aging list.

### Keep slot = Yes

indicates the auxiliary storage address (ASA) to which a page is written during frame replenishment will remain allocated to that page when it is read into real memory upon access by the guest.

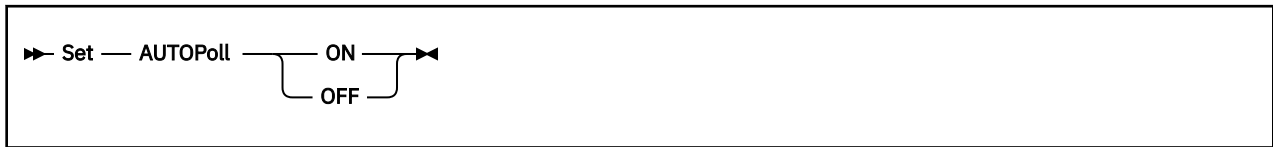
### Keep slot = No

indicates the ASA to which a page is written during frame replenishment is unallocated during page fault processing.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*|command contains extra option(s) starting with *option*
- HCP6704E Missing token at end of line

## SET AUTOPOLL



### Authorization

Privilege Class: G

### Purpose

Use SET AUTOPOLL to control how CP detects changes to virtual autopolling channel programs:

- The system itself tests for the change
- Your virtual machine notifies the system of the change.

**Note:** In general, the guest operating system issues the notifying command without your intervention.

### Operands

**ON**

**OFF**

controls whether CP tests for changes to virtual autopolling channel programs. ON indicates that your virtual machine must notify CP when a change is made to a virtual autopolling channel program. OFF indicates that CP tests for changes to virtual autopolling channel programs.

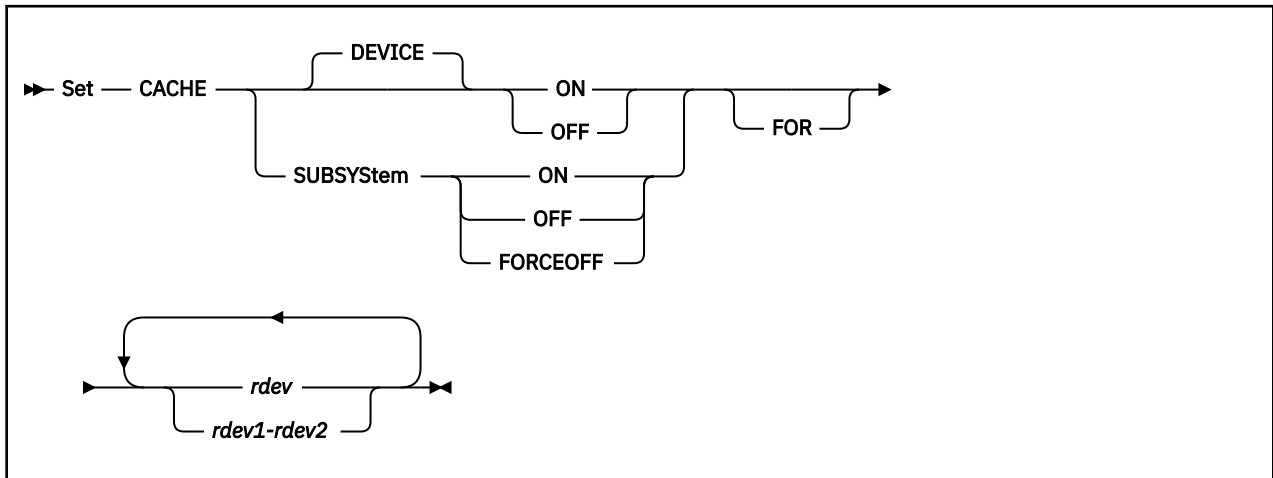
### Usage Notes

1. Do not enter this command arbitrarily. If you enter the command while a guest operating system is running, you may cause the system to stop or to end abnormally.
2. If SET AUTOPOLL ON is required for your virtual machine, the operating system running in your virtual machine is using DIAGNOSE X'28' to notify the system whenever a virtual autopolling channel program has been modified.
3. SET AUTOPOLL OFF is required if the operating system running in your virtual machine does NOT use the DIAGNOSE X'28' interface.

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP263E Too many operands were supplied for this command.

## SET CACHE



### Authorization

Privilege Class: B

### Purpose

Use SET CACHE to activate or deactivate the cache function by device or by subsystem. The subsystems are IBM DASD subsystems.

### Operands

#### DEVICE ON

indicates that caching is to be activated for the specified devices. DEVICE is the default.

#### DEVICE OFF

indicates that caching is to be deactivated for the specified devices. Any user to whom the device was attached bypasses the cache.

#### SUBSYStem ON

indicates that caching is to be made available for the entire subsystem to which the device is attached.

#### SUBSYStem OFF

indicates that caching is to be made unavailable for the entire subsystem to which the device is attached.

#### SUBSYStem FORCEOFF

indicates that the cache is to be forced to be unavailable to subsystems to which the devices are attached.

*rdev*

*rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers of devices in the subsystem.

### Usage Notes

1. All SET CACHE operands are nonpositional except that if FOR is specified, it must precede the device list.
2. Device caching and subsystem cache availability are both required for the caching function to be used, but they are controlled independently. Caching can be turned on for a device, but unless the

subsystem cache is available, caching does not occur. The subsystem remembers the caching state of each device even when the subsystem cache is not available.

3. The caching function is always enabled on RAID (Redundant Array of Independent Disks) technology DASD such as an IBM DASD subsystem. For these types of DASD, the device accepts but ignores this command; the caching function cannot be modified by the user. The caching function for these types will always appear as ON in related query commands.
4. Error message HCP296E appears if the subsystem is not in the correct mode to accept the QUERY CACHE command.
5. SET CACHE SUBSYSTEM FORCEOFF is a recovery command. It can be issued only after a SET CACHE SUBSYSTEM OFF has been issued and the subsystem could not make the cache unavailable because of internal errors.
6. If SET CACHE SUBSYSTEM FORCEOFF is issued, the subsystem causes all current channel programs operating on caching devices to fail.
7. SET CACHE SUBSYSTEM OFF and SET CACHE SUBSYSTEM FORCEOFF halt, but do not end, Concurrent Copy (CONCOPY) sessions on the subsystem. Because the CONCOPY function requires that caching be available for the subsystem, applications that had active CONCOPY sessions on the subsystem must end and then restart the CONCOPY function after a SET CACHE SUBSYSTEM OFF or SET CACHE SUBSYSTEM FORCEOFF command has been issued.
8. This command may produce delayed responses. See [“Delayed Responses to CP Commands”](#) on page 7 for more information.
9. This command may be issued to offline devices.
10. When specifying a real device number (*rdev*), do not use the secondary device in a duplex pair. The QUERY DUPLEX and QUERY DASD DETAILS commands can be issued to determine if a device is a secondary device in a duplex pair.
11. Error message HCP263E appears if too many devices are specified.

### Examples

To activate device 340 for caching, enter SET CACHE DEVICE ON 340.

### Responses

Response 1:

```
rdev CACHE activated for device
```

If the originator of the command is not the system operator, the following response is issued to the system operator and is also repeated for each device specified on the command line.

```
rdev CACHE was activated for device by userid.
```

#### ***rdev***

is the real device number of the device that had been given access to the cache.

#### ***userid***

is the ID of the user who issued the command.

Response 2:

**Immediate Response:** If the cache was deactivated immediately, the following response is displayed for each specified device when you enter SET CACHE DEVICE OFF:

```
rdev CACHE deactivated for device
```

**Delayed Response:** If the cache is not deactivated immediately, the following response is displayed when you enter SET CACHE DEVICE OFF:

```
Command started: SET CACHE DEVICE OFF rdev.
```

## SET CACHE

When the cache is deactivated, the following response is issued:

```
rdev CACHE deactivated for device
```

Under certain circumstances, it is possible that a cache storage subsystem does not notify CP when it has completed deactivating the cache. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing ends after generating the following response:

```
Command results lost: SET CACHE DEVICE OFF rdev.
```

In this case the QUERY CACHE command can be issued to determine the state of the cache.

If errors occur during the asynchronous processing of the command which prevent the successful completion of the command, the following response is displayed:

```
Command failed: SET CACHE DEVICE OFF rdev.
```

### **rdev**

is the real device number of a specified device for which the subsystem cache is available.

**Response to System Operator:** If the originator of the command is not the system operator, the following response is issued to the system operator and is also repeated for each device specified on the command line.

```
rdev CACHE was deactivated for device by userid.
```

### **rdev**

is the real device number of a device for which caching has been deactivated.

### **userid**

is the ID of the user who issued the command.

Response 3:

**Immediate Response:** If the subsystem cache is made available immediately, the following response is repeated for each *unique subsystem* that is associated with the devices specified when you enter SET CACHE SUBSYSTEM ON:

```
rdev CACHE available for subsystem
```

### **rdev**

is the real device number of a specified device for which the subsystem cache has been activated.

**Delayed Response:** If the cache is not made available immediately, the following response is displayed when you enter SET CACHE SUBSYSTEM ON:

```
Command started: SET CACHE SUBSYSTEM ON rdev.
```

When the cache is made available, the following responses are issued:

```
rdev CACHE available for subsystem
```

Under certain circumstances, it is possible that a cache storage subsystem does not notify CP when it has completed making the cache available. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing is ended after generating the following response:

```
Command results lost: SET CACHE SUBSYSTEM ON rdev.
```

In this case, the QUERY CACHE command can be issued to determine the state of the cache.

If errors occur during the asynchronous processing of the command preventing the successful completion of the command, the following response is displayed:



```
Command failed: SET CACHE SUBSYSTEM ON rdev.
```

***rdev***

is the real device number of a specified device for which the subsystem cache has been made available.

**Response to System Operator:** If the originator of the command is not the system operator, the following response is issued to the system operator when the subsystem cache has been made available. This response is also repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev CACHE was made available for subsystem by userid.
```

***rdev***

is the real device number of one of the specified devices in a subsystem for which the cache has been made available.

***userid***

is the ID of the user who issued the command.

Response 4:

**Immediate Response:** If you enter SET CACHE SUBSYSTEM FORCEOFF, or if the cache is made unavailable immediately when you enter SET CACHE SUBSYSTEM OFF, the following response is repeated for each *unique subsystem* that is associated with the devices specified.

```
rdev CACHE unavailable for subsystem
```

***rdev***

is the real device number of a specified device for which the subsystem cache has been made unavailable.

**Delayed Response:** If the cache is not made unavailable immediately, the following response is displayed when you enter SET CACHE SUBSYSTEM OFF:

```
Command started: SET CACHE SUBSYSTEM OFF rdev.
```

When the cache is made unavailable, the following responses are issued:

```
rdev CACHE unavailable for subsystem
```

Under certain circumstances, it is possible that a cache storage subsystem does not notify CP when it has completed making the cache unavailable. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing ends after generating the following response:

```
Command results lost: SET CACHE SUBSYSTEM OFF rdev.
```

In this case, the QUERY CACHE command can be issued to determine the state of the cache.

If errors occur during the asynchronous processing of the command that prevent the successful completion of the command, the following response is displayed:

```
Command failed: SET CACHE SUBSYSTEM OFF rdev.
```

***rdev***

is the real device number of a specified device for which the subsystem cache has been made unavailable.

**Response to System Operator:** If the originator of the command is not the system operator, the following response is issued to the system operator when the subsystem cache has been made unavailable. This

## SET CACHE

response is also repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev CACHE was made unavailable for subsystem by userid.
```

### ***rdev***

is the real device number of one of the specified devices in a subsystem for which the cache has been made unavailable.

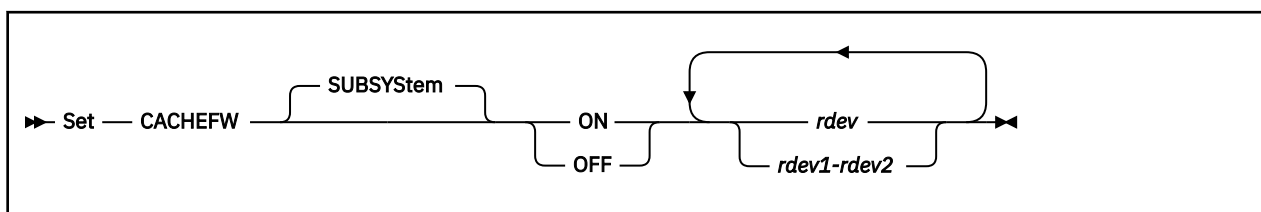
### ***userid***

is the ID of the user who issued the command.

## Messages

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP009E Invalid range - *range*
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP263E Too many operands were specified for this command.
- HCP296E Status is not as required - *rdev; text*
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*
- HCP1109E The *command* command cannot be processed for device *rdev* because the device is the secondary in a duplex pair.
- HCP1120E The requested [command|DETACH|GIVE] for device [*ldev|rdev*] did not complete in the allotted time.

## SET CACHEFW



### Authorization

Privilege Class: B

### Purpose

Use the SET CACHEFW command to activate or deactivate the cache fast write function in a DASD subsystem. This command is valid for cache storage subsystems that support the cache fast write function.

### Operands

#### SUBSYSstem

specifies that the entire storage subsystem is affected by this command, regardless of which device within the subsystem receives the order. This is the default (and only) value.

#### ON

indicates that the cache fast write function is to be activated for the subsystem.

#### OFF

indicates that the cache fast write function is to be deactivated for the subsystem.

#### rdev

#### rdev1-rdev2

is the real device number, a list of real device numbers, or a range of real device numbers of devices in the subsystems.

### Usage Notes

1. The SET CACHEFW function is only valid for storage subsystems that support the cache fast write function.
2. If the subsystem cache is unavailable when the SET CACHEFW ON command is issued, cache fast write data access begins when the cache is made available (through SET CACHE SUBSYSTEM ON).
3. When SET CACHEFW OFF is issued, the cache fast write data for all devices in the subsystem is destaged to the appropriate DASD.
4. There are certain error messages that may result if the subsystem is not in the correct mode to accept the SET CACHEFW ON command. See error message HCP296E for more information.
5. SET CACHEFW OFF may produce delayed responses. See [“Delayed Responses to CP Commands” on page 7](#) for more information.

### Responses

Response 1:

## SET CACHEFW

The following response is displayed when you enter SET CACHEFW ON and is repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev Cache fast write function has been activated.
```

If the originator of the command is not the system operator, the following response is issued to the system operator and is also repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev Cache fast write function was activated by userid.
```

### **rdev**

is the real device number of one of the specified devices in a subsystem for which the cache fast write function has been activated.

### **userid**

is the ID of the user who issued the command.

Response 2:

**Immediate Response:** If you enter the SET CACHEFW OFF command and the cache fast write function is deactivated immediately, the following response is displayed. This response is repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev Cache fast write function has been deactivated.
```

### **rdev**

is the real device number of one of the specified devices in a subsystem for which the cache fast write function has been deactivated.

**Delayed Response:** If the cache fast write function is not deactivated immediately, the following response is issued:

```
Command started: SET CACHEFW OFF rdev.
```

When the cache fast write function is deactivated, the following responses are issued:

```
rdev Cache fast write function has been deactivated.
```

Under certain circumstances, it is possible that a cache storage subsystem does not notify CP when it has deactivated the cache fast write function. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing is ended after generating the following response:

```
Command results lost: SET CACHEFW OFF rdev.
```

In this case, the QUERY CACHEFW command can be issued to determine the state of the cache fast write function.

If errors occur during the asynchronous processing of the command that prevent the successful completion of the command, the following response is displayed:

```
Command failed: SET CACHEFW OFF rdev.
```

### **rdev**

is the real device number of one of the specified devices in a subsystem for which the cache fast write function has been deactivated.

**Response to System Operator:** If the originator of the command is not the system operator, the following response is issued to the system operator when the cache fast write function has been deactivated. This

response is also repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev Cache fast write function was deactivated by userid.
```

***rdev***

is the real device number of one of the specified devices in a subsystem for which the cache fast write function has been deactivated.

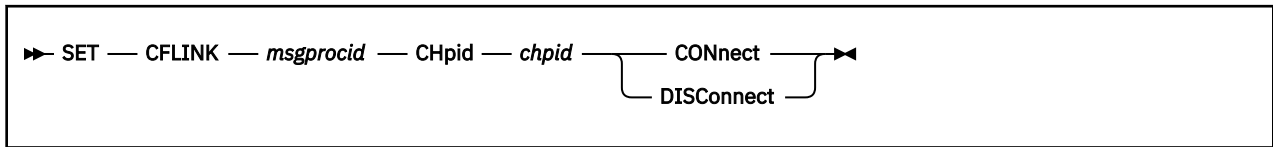
***userid***

is the ID of the user who issued the command.

**Messages**

- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP009E Invalid range - *range*
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP319E Error processing subsystem command - *cmd* for DASD *rdev*
- HCP332E Invalid control unit type - *rdev*
- HCP1109E The *command* command cannot be processed for device *rdev* because the device is the secondary in a duplex pair.

## SET CFLINK



### Authorization

Privilege Class: A, B, G

### Purpose

The SET CFLINK command logically connects or disconnects a coupling facility link using the message processor user ID and channel path ID to identify the link.

### Operands

#### *msgprocid*

is the user ID of the coupling facility virtual machine that owns the link.

#### **CHPID** *chpid*

Specifies the channel path ID of the link, which is known by the coupling facility virtual machine that owns the link.

#### **CONNect**

Enable and logically connect the link.

#### **DISConnect**

Disable and logically disconnect the link.

### Usage Notes

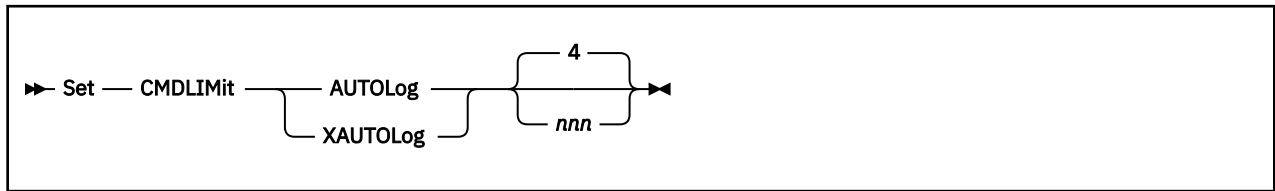
1. The issuing user must either be class A or class B. If class G only, the user must be coupled to the indicated coupling facility virtual machine.
2. The user ID specified by *msgprocid* must be logged on and must be a coupling facility virtual machine.
3. For the CONNect operand, the link specified by *chpid* must already exist. If it is already connected, no action is taken.
4. For the DISCONNect operand, the link specified by *chpid* must already exist. If it is already disconnected, no action is taken.

### Messages

- HCP001E Unknown CP command: *command*
- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP045E *userid* not logged on
- HCP047E CFLINK with this CHPID does not exist
- HCP260E *command* command processing cannot complete due to errors.
- HCP846E Invalid channel path identifier.
- HCP1014E A required option is missing.

- HCP2800E *msgprocid* is not a CF Service Machine.
- HCP2805E One or more message processors does not exist in your I/O configuration.
- HCP2806E You are not allowed to control message processors.
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*

## SET CMDLIMIT



### Authorization

Privilege Class: B

### Purpose

Use SET CMDLIMIT to limit the number of incorrect passwords to be processed with the AUTOLOG and XAUTOLOG commands. When a user exceeds this limit, CP disables the commands for the rest of the user logon session.

### Operands

**AUTOLOG *nnn***

**XAUTOLOG *nnn***

sets the system limit for incorrect passwords entered on the AUTOLOG or XAUTOLOG command. The maximum limit is 255, the minimum limit is 1, and the default limit is 4.

### Usage Notes

1. For security reasons, CP limits the number of incorrect passwords allowed on the AUTOLOG and XAUTOLOG commands in each user's logon session. The SET CMDLIMIT lets you raise or lower that limit.
2. Use the QUERY CMDLIMIT command to determine the current system limit value.
3. AUTOLOG is an accepted synonym for XAUTOLOG. The minimum truncation is AUTOL.
4. The initial system setting depends on how the JOURNALING system configuration statement is specified at system generation. If the FACILITY ON operand is specified, and no value is specified for the number of logon attempts, the initial setting of the limit is 4.
5. If a user exceeds the value specified by *nnn* and logon journaling is enabled, the user will not be able to enter the AUTOLOG or XAUTOLOG commands as long as logon journaling is enabled, even if the CMDLIMIT value *nnn* is increased with the SET CMDLIMIT command.

If a user exceeds the value specified by *nnn* and logon journaling is not enabled, the user will not be able to enter the AUTOLOG or XAUTOLOG commands. However, if the value *nnn* is increased with the SET CMDLIMIT command, the user will be allowed additional attempts to use the AUTOLOG or XAUTOLOG commands.

If a user exceeds the value specified by *nnn* and logon journaling is enabled (and is prevented from executing the AUTOLOG or XAUTOLOG commands) and then the SET JOURNAL LOGON OFF command is issued, the number of invalid password attempts accrued when logon journaling was enabled will not be reset. Consequently, if the *nnn* value of SET CMDLIMIT is not increased, the user will still not be able to run the AUTOLOG or XAUTOLOG commands. However, if the *nnn* value of SET CMDLIMIT is increased, the user will be allowed additional attempts to use the AUTOLOG or XAUTOLOG commands.

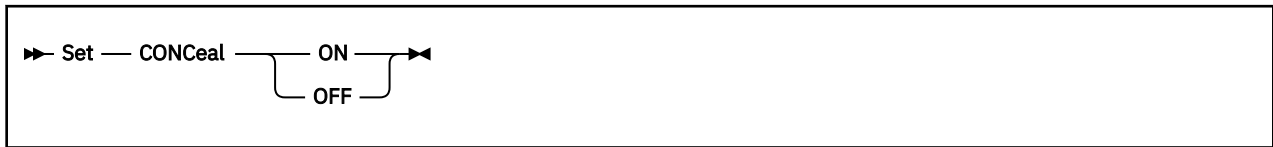
### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}



- HCP1014E A required option is missing.
- HCP1826E The system limit of incorrect passwords was specified with an invalid number - *number*

## SET CONCEAL



### Authorization

Privilege Class: G

### Purpose

Use SET CONCEAL to turn on and off the protected application environment. This environment protects a session from unexpectedly entering CP READ.

### Operands

#### ON

specifies that the virtual machine is to operate in the protected application environment. This environment is intended for the application end user who does not want to interact with CP. Instead of presenting the user with an unexpected CP READ on occurrence of certain error conditions (for example, paging error, soft abend, disabled wait PSW loaded), the protected application facility forces an automatic re-IPL of the virtual machine without issuing any error messages to the user. The IPL is initiated using the IPL command last used for the virtual machine.

To prevent a re-IPL loop, CP does not force re-IPL if a previous automatic re-IPL occurred less than a minute before. No more than 10 automatic re-IPLs are attempted between activation and deactivation of the protected application environment. If a re-IPL is not attempted or if the attempt fails, CP issues the same message as when the protected application environment is not active and puts the terminal into CP READ state. For 3270 terminals, SET CONCEAL ON disables the break key. If you subsequently deactivate the protected application environment using the SET CONCEAL OFF command, the break key and the PA1 key are reenabled. You may also enable the break key by issuing a TERMINAL BRKKEY command while in the protected application environment.

#### OFF

specifies that the virtual machine should no longer operate in the protected application environment. Certain error conditions (such as a paging error, a soft abend, or a disabled wait PSW in the virtual processor) cause CP to take control of the virtual machine and display CP READ on the terminal. No re-IPL is initiated. The terminal break key, PA1 key, and ASCII double break or attention are reenabled when you enter SET CONCEAL OFF.

### Usage Notes

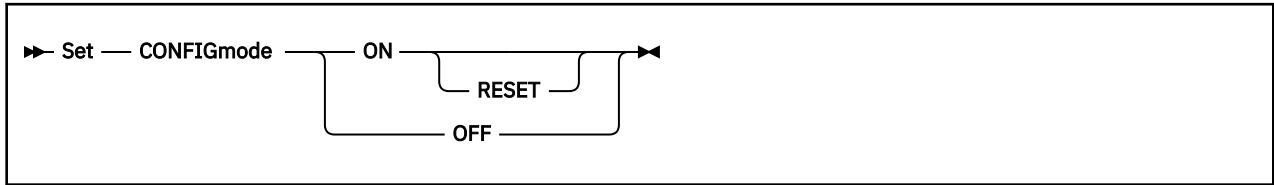
1. With your break key disabled, you cannot enter the CP READ state while running in full screen mode unless your guest (operating system or application) provides a way. The guest can use DIAGNOSE X'08' to pass either the CP command or no command at all to enter CP READ. In addition, the TERMINAL MODE command allows entry to CP on a console attention interrupt when in line mode.
2. An application can establish the protected application environment for itself by issuing a SET CONCEAL ON command using DIAGNOSE X'08'.
3. You may not specify SET CONCEAL ON if you have defined a multiprocessor configuration. The protected application facility only supports virtual uniprocessor systems.
4. If you use an application that runs with the protected application environment active, be aware that an automatic re-IPL resets your virtual machine. If you IPL a named saved system, storage is also cleared; therefore, data in virtual machine storage is lost.

5. If the protected application environment is already active, the SET CONCEAL ON command has no effect. Thus, if the break key has been reenabled using the TERMINAL BRKKEY command, this second SET CONCEAL ON command does not disable the break key.
6. For ASCII devices, hitting a double break or attention while in the protected application environment posts an attention to the virtual machine rather than presenting a CP READ. The only way to enable the double break or attention again is to SET CONCEAL OFF.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP5780E Adjunct does not support the *command subcommand* command
- HCP6026E Operand missing or invalid.

## SET CONFIGMODE



### Authorization

Privilege Class: B

### Purpose

Use SET CONFIGMODE to turn configuration mode on or off. Configuration mode is a machine state that is used to change your I/O configuration dynamically. You can use this command to issue a group of dynamic I/O commands together. Otherwise, each command enters and exits configuration mode at the beginning and ending of each dynamic I/O command.

Note that the SET DYNAMIC\_I/O OFF and SET LSYSTEM commands are an exception to the above. See Usage Note “3” on page 1437.

The dynamic I/O commands are:

DEFINE CHPID	DEFINE PATH
DELETE CHPID	DELETE PATH
MODIFY CHPID	MODIFY PATH
DEFINE CU	DEFINE CNTLUNIT
DELETE CU	DELETE CNTLUNIT
MODIFY CU	MODIFY CNTLUNIT
DEFINE DEVICE	DEFINE IODEVICE
DELETE DEVICE	DELETE IODEVICE
MODIFY DEVICE	MODIFY IODEVICE
DEFINE LPAR	
DELETE LPAR	
SETCONFIGMODE	SET DYNAMIC_I/O
SET LSYSTEM	

### Operands

#### ON

tells CP to enter configuration mode.

#### OFF

tells CP to exit configuration mode.

#### RESET

tells CP to enter or exit configuration mode even if there is a mismatch between the system's and the hardware's view of the I/O configuration.

### Usage Notes

1. Whenever you issue one of the dynamic I/O commands, CP puts you in configuration mode, processes the dynamic I/O command, and then takes you out of configuration mode.
2. Whenever you dynamically change your I/O configuration using one of the dynamic I/O commands, CP must perform a certain amount of overhead. If you want to issue several dynamic I/O commands in succession and you are concerned about performance, issue the SET CONFIGMODE ON command,

issue your dynamic I/O commands, and then issue the SET CONFIGMODE OFF command. This method causes CP to perform the overhead once at the beginning and once at the end, rather than before and after each individual command.

3. There are two dynamic I/O commands that you cannot issue while in configuration mode: the SET DYNAMIC\_I/O OFF command and the SET LSYSTEM command. You must issue SET DYNAMIC\_I/O OFF and SET LSYSTEM as individual commands. For more information about the SET DYNAMIC\_I/O command, see [“SET DYNAMIC\\_I/O” on page 1502](#). For more information about the SET LSYSTEM command, see [“SET LSYSTEM” on page 1553](#).
4. When you use the dynamic I/O commands, you are changing the I/O configuration as the software (CP) understands it. You must also change the I/O configuration as the hardware understands it. To do so, you must update the IOCP source file and issue the SET IOCDs\_ACTIVE command (see [“SET IOCDs\\_ACTIVE” on page 1525](#)).
5. If you are in configuration mode and you issue a dynamic I/O command that does not complete successfully, CP automatically takes you out of configuration mode.
6. If you are in configuration mode and any of the following occur:
  - The system abends,
  - The system loses power,
  - You issue a SHUTDOWN REIPL command,
  - You issue a CLEAR RESET from the hardware console, or
  - You issue a LOAD CLEAR from the hardware console.

CP takes you out of configuration mode before shutting down the system.
7. To display whether the processor is currently in configuration mode, use the QUERY CONFIGMODE command (see [“QUERY CONFIGMODE” on page 806](#)).

## Responses

### Response 1:

To turn configuration mode on, enter the following SET CONFIGMODE command:

```
set configmode on
System is now in configuration mode
Ready;
```

If configuration mode is on when you issue the SET CONFIGMODE ON command, you see the following response:

```
set configmode on
System is already in configuration mode
Ready;
```

### Response 2:

To turn configuration mode off, enter the following SET CONFIGMODE command:

```
set configmode off
System is no longer in configuration mode
Ready;
```

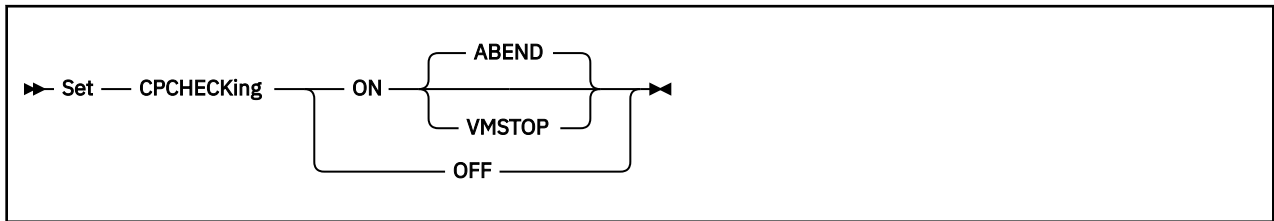
If configuration mode is off when you issue the SET CONFIGMODE OFF command, you see the following response:

```
set configmode off
System is not in configuration mode
Ready;
```

### Messages

- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request [for channel path|for control unit|for device] failed with return code *nnnn*
- HCP6808E Cannot find the configuration token on the system
- HCP6808E System disabled for dynamic I/O changes
- HCP6808E The configuration mode setting of the hardware and software do not match
- HCP6808E The dynamic I/O support was disabled at the service console
- HCP6808E The VM configuration token does not match the hardware configuration token
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Cannot find configuration token in the system
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed on this system
- HCP6819E A dynamic I/O command is currently executing for user *userid*
- HCP6851E I/O configuration cannot be changed because VM is not the controlling partition

## SET CPCHECKING



### Authorization

Privilege Class: A, C

### Purpose

Use SET CPCHECKING to activate or deactivate CP internal checking.

#### ON

indicates that internal CP checking should be activated. The action to be taken when an untrue assertion is encountered is controlled by the subsequent parameters.

#### ABEND

indicates that an abend should occur. Whether it is a hard abend or a soft abend is controlled by the specific assertion case. It is typically a hard abend. This is the default when CP internal checking is activated.

#### VMSTOP

this parameter is permitted only when CP is itself running in a virtual machine, and is generally useful only for debugging CP. Specifying this parameter causes CP to issue a DIAGNOSE X'08' instruction when an untrue assertion is encountered, specifying a command string length of zero. This causes the virtual machine to stop and for CP to post a read to the console. Messages are sent to the console (from a DIAGNOSE X'08' MESSAGE \* command) providing information about the cause of the stop.

#### OFF

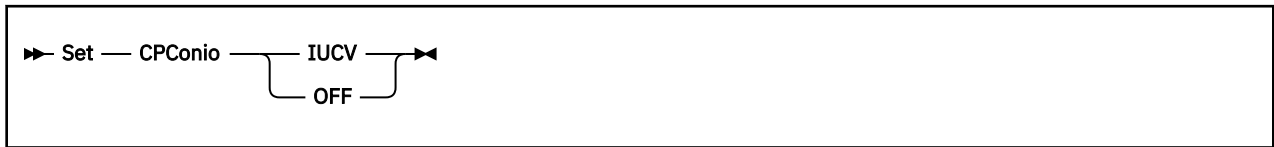
Indicates that CP internal checking should be deactivated.

### Messages

- HCP1007E The VMSTOP option is only permitted when CP itself is running in a virtual machine.
- HCP6010I The CPCHECKING setting has been changed to OFF to avoid a possible abend loop. It may be set ON again if desired.

## SET CPCONIO

---



### Authorization

Privilege Class: G

### Purpose

Use SET CPCONIO to control the destination of all CP output. At LOGON, CPCONIO is set OFF.

### Operands

#### IUCV

specifies that responses are to be passed to the virtual machine through IUCV if a connection to the message system service exists. If the console is being spooled, the responses are also placed in the console spool file, as well as being sent through IUCV. If no IUCV connection exists, data is handled as if SET CPCONIO OFF had been issued.

#### OFF

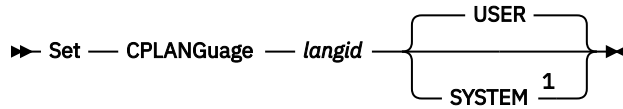
specifies that responses are displayed at the terminal.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid



## SET CPLANGUAGE



Notes:

<sup>1</sup> SYSTEM is valid only for class B.

### Authorization

Privilege Class: B, G

### Purpose

Use SET CPLANGUAGE to change the language of your virtual machine or the entire system.

### Operands

#### *langid*

is the language to which the virtual machine is to be set; *langid* must be 1 to 5 characters long and must consist of CMS file system characters only.

#### **USER**

specifies that the default language is being set for the user.

#### **SYSTEM**

specifies that the default language is being set for the entire system.

### Usage Notes

1. Be aware of the language-related terminal restrictions when you specify a language. If you are not sure what characters your terminal can display, check with your system administrator.
2. The QUERY CPLANGLIST command displays all the valid language IDs that you can set in CP. To display the language that is currently active for CP in your virtual machine, enter the QUERY CPLANGUAGE command. Contact your system administrator if you have any questions about the languages available on your system.
3. The system default language is set with the CPLANGUAGE operand of the USER\_DEFAULTS system configuration statement or with the CP SET CPLANGUAGE command. The order of precedence, from highest priority to lowest, that determines an individual user's language is:
  - SET CPLANGUAGE command specified by the individual user
  - the value specified for the LANG operand of the OPTION statement in the user's directory
  - the system default language, and then
  - the language that is built into the CP nucleus.
4. If the system default language is changed, the response to QUERY CPLANGLIST will show the new system default language. If the user is using the system default language, the user's current language will not change until the user either logs off and logs back on, or issues the CP SET CPLANGUAGE command or CMS SET LANGUAGE command to switch to the new system default language.
5. CP uses the message repository that is built into the CP nucleus even when a message repository file for the same language exists. The QUERY NLS command can be used to display the message repository files that exist on the system.

## SET CPLANGUAGE

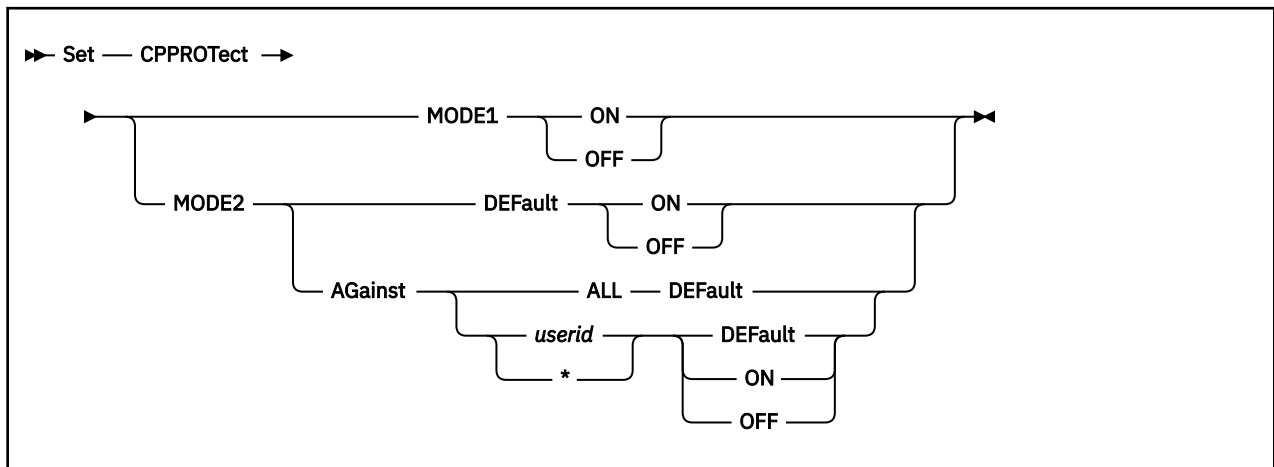
### Examples

If your virtual machine is set to American English for CP, and you want to change your language to German, enter `set cplanguage ger`. To change it back to American English, enter `set cplanguage ameng`.

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP1831E The command could not be completed because of a temporary condition. Try again later.
- HCP2575E The message repository for *langid* cannot be found.
- HCP2576E The command processing cannot complete and a soft abend has been taken because an unrecoverable system error has occurred.
- HCP2577E Message repository file *name* is damaged
- HCP6525E External Security Manager is unavailable

## SET CPPROTECT



### Authorization

Privilege Class: C

### Purpose

Use SET CPPROTECT to control enablement of certain security modes in z/VM, such as limiting speculative execution in the machine. Speculative execution is an optimization mechanism designed into the processor to improve system performance.

The standard virtual machine isolation and protection mechanisms in z/VM are designed to prevent code being run in an unprivileged virtual machine from interfering with or gaining unauthorized access to resources or data owned by the z/VM Control Program or by other guests. Setting CPPROTECT modes ON offers additional protections against information leak attacks that might be attempted by malicious or compromised code running in an untrusted virtual machine. Running the system with MODE1 ON or MODE2 ON might increase system overhead, reduce capacity, or reduce throughput.

z/VM maintains a system-wide MODE1 setting and a MODE2 default setting. When z/VM is IPLed, MODE1 is ON and the default for MODE2 is OFF. In addition to changing the MODE1 setting and the default MODE2 setting, SET CPPROTECT can also establish per-user MODE2 override settings.

In choosing CPPROTECT settings, the system programmer might want to consider the degree to which the code running in virtual machines is known and controlled, the level of risk from unauthorized discovery of data in the memory of CP or other virtual machines, and the performance cost associated with the various protections. For example, an open multitenant cloud platform might call for greater protection than an isolated internal company system.

Details on the protection modes are not provided in this publication. Authorized personnel of IBM clients can consult the IBM Z Security Portal for further information and guidance. To request access to the portal, go to [Register for IBM Z Security Portal \(www.ibm.com/systems/campaignmail/z/capabilities/system-integrity/register-zsecurity-portal\)](http://www.ibm.com/systems/campaignmail/z/capabilities/system-integrity/register-zsecurity-portal). Scroll and review information on that page including the Terms and Conditions section. The IBM Z Security Portal section of the page includes links to request access and to obtain the required IBMid.

### Operands

#### MODE1

refers to a system-wide mitigation mechanism that is either ON or OFF.

**MODE2**

refers to a mitigation mechanism that the z/VM Control Program (CP) can apply when it interacts with some users but not with others. MODE2 protection can be enabled or disabled against all users or against specific users.

**DEfault**

specifies that the system-wide default is to be set.

**AGainst ALL DEfault**

specifies that every logged-on user will be set to the default MODE2 setting. All user overrides will be removed.

**AGainst userid****AGainst \***

specifies that a user-specific override is to be set, taking precedence over the system default when the specified virtual machine is run. An asterisk refers to the virtual machine of the command issuer.

**AGainst userid DEfault**

cancels the user-specific override for the specified user. Subsequent handling of this virtual machine will be governed by the system-wide default MODE2 setting.

**ON****OFF**

indicates that a level of protection should be enabled (ON) or disabled (OFF) under the specified conditions.

**Usage Notes**

1. At z/VM IPL, the MODE1 setting is ON and the default MODE2 setting is OFF. These settings apply to all users on the system. To change these IPL settings, insert the following statement(s) into the directory of the primary system operator user ID (usually OPERATOR):

```
COMMAND SET CPPROTECT MODE1 OFF
COMMAND SET CPPROTECT MODE2 DEFAULT ON
```

Because the operator is the first user automatically logged on at IPL, this will put the setting into effect before any virtual machine can run.

2. At logon, each user receives the system default MODE2 setting.
3. A user-specific MODE2 override remains in effect until the user logs off. To establish a persistent override for a user, insert the following statement into either the USER or IDENTITY directory entry, or the PROFILE entry included by this entry:

```
COMMAND SET CPPROTECT MODE2 AGAINST * {ON|OFF}
```

This will ensure that the setting is put into effect as soon as the user logs on.

4. If APAR VM65414 or its precursor, VM65396, is installed on the source and destination systems, VMRELOCATE will preserve a user-specific override set for the relocated virtual machine. If either APAR is installed on the source system but neither is installed on the destination system, a user override will be lost in relocation. If either APAR is installed on the destination system but neither is installed on the source system, the relocated guest will inherit the default MODE2 setting on the destination system.
5. Machine support is required to implement MODE1 ON and MODE2 ON. If the support is not available, the requested ON setting will be recorded and will be preserved by VMRELOCATE as described above but will not be effective on this system. If the necessary machine service is applied dynamically, you can then issue QUERY CPPROTECT to cause CP to recognize the machine support and put the requested settings into effect.

**Examples**

Assuming a freshly IPLed system with the required machine support and no CP directory COMMAND statements, consider the following sequence of commands:

```
set cprotect mode1 on
set cprotect mode2 default off
set cprotect mode2 against alice on
set cprotect mode2 against bob off
```

MODE1 will be ON system-wide. MODE2 will be OFF with respect to all users except ALICE: with respect to BOB because of the user-specific override, and with respect to all others because of the system default.

If the following command is subsequently issued:

```
set cprotect mode2 default on
```

MODE2 remains OFF with respect to user BOB, because of the user-specific override. MODE2 remains ON to protect against ALICE, because of the user-specific override. The rest of the users now run with MODE2 ON, because of the changed system default.

To remove all user overrides, in other words ALICE's and BOB's overrides:

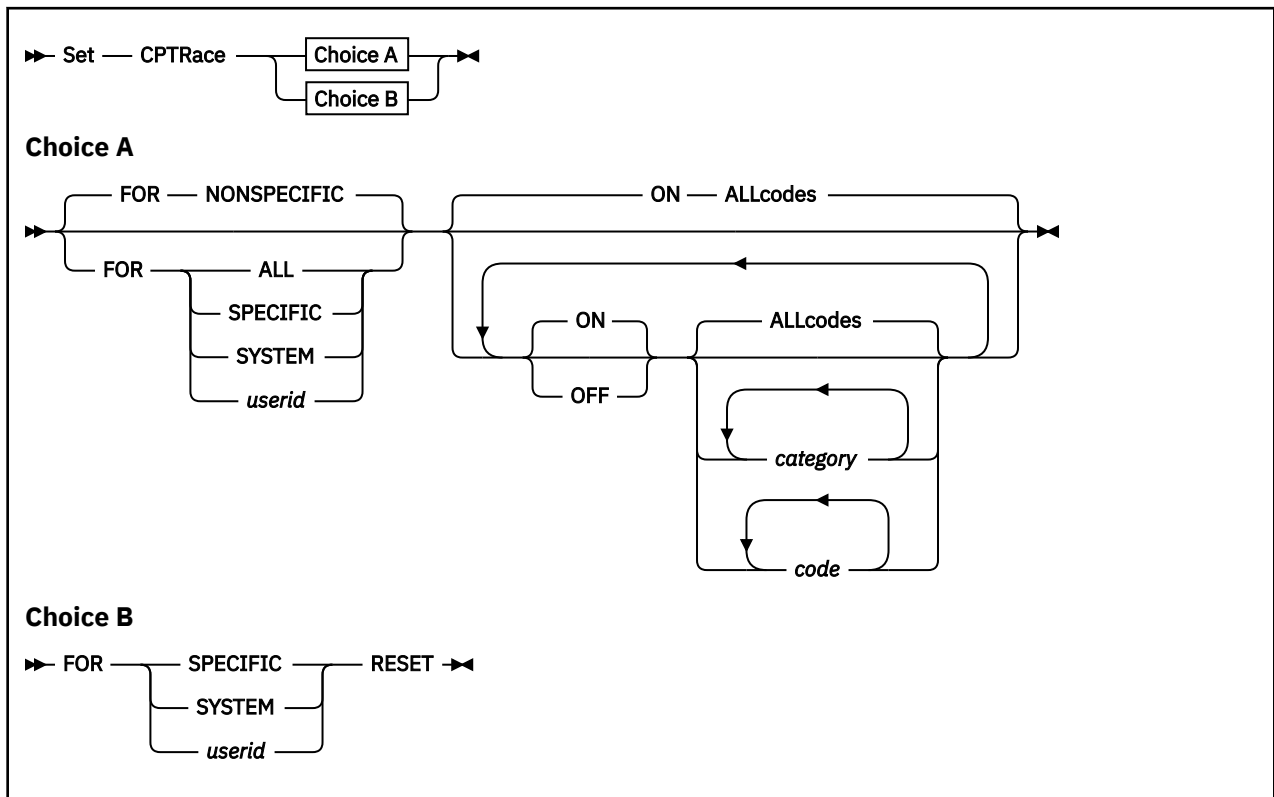
```
set cprotect mode2 against all default
```

The user-specific overrides for ALICE and BOB are removed. All users now run with the system default, which in this example is MODE2 ON.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP1056I {MODE1|MODE2} ON request set, but required machine support not available.

## SET CPTRACE



### Authorization

Privilege Class: A, C

### Purpose

Use SET CPTRACE to activate or deactivate the trace function for real machine events. (Trace information is logged in trace tables.)

### Operands

#### CPTRACE

enables or disables tracing of real machine events in the CP trace tables.

#### NONSPECIFIC

indicates that the specified tracing applies to all virtual machines not specifically set or named by a previous SET CPTRACE command. This is the default. All virtual machines begin in the NONSPECIFIC group. All members of the NONSPECIFIC group have the same codes traced.

#### SPECIFIC

#### SYSTEM

#### ALL

#### *userid*

specifies the target for the subsequent tracing changes.

SPECIFIC indicates that the specified tracing applies only to those virtual machines specifically set or named by a previous SET CPTRACE command. Members of the SPECIFIC group may each have a unique set of codes traced.

SYSTEM indicates that the specified tracing applies to system work. SYSTEM becomes a member of the SPECIFIC group, even though the specified tracing may not match the virtual machines already in the group. Therefore, SYSTEM is similar to *userid* in that it names the work for which tracing applies, and SYSTEM can also be a part of either the SPECIFIC or NONSPECIFIC groups, but not both.

ALL indicates that the specified tracing applies to both the SPECIFIC and NONSPECIFIC groups of virtual machines and to system work.

*userid* indicates that the specified tracing applies only to this specific virtual machine. The operand *userid* becomes a member of the SPECIFIC group, even though the specified tracing may not match others in the group.

**ON****OFF**

enables or disables tracing of the specified trace categories or codes. ON is the default. ON and OFF may be intermixed in the same trace category list to enable and disable trace categories in the same command. The trace category list is processed from left to right.

**ALLcodes**

sets or resets tracing of all events. This is the default.

**category  
code**

is either a single 4-digit hexadecimal code for an event to be traced or the name of a category of trace codes. The system displays only the setting of this code or category.

The category names, subcategory names, and the trace codes in each category are listed below in system hierarchical order. For a full numeric list of trace codes regardless of category, refer to [Trace Table Codes](#) in *z/VM: Diagnosis Guide*.

**EXT**

External interrupt activities:

**PCDIAG**

Processor controller (PC) diagnose:

**3C00**

PC diagnose request started

**3C55**

PC diagnose request returned

**PCSERVC**

Processor controller (PC) service call:

**3C10**

PC service call request started

**3C65**

PC service call request returned

**3CFF**

Unsolicited processor controller interrupt received

**8100**

External interrupt (z/Architecture)

**AE01**

SIGP instruction

**SVC**

Supervisor call interrupt related activities:

**0200**

SVC interrupt

**2200**

Add or stack a COMBK

## SET CPTRACE

### **2301**

Execute CP command

### **CALLRET**

Call return trace codes:

#### **2800**

Call-with-save-area

#### **2810**

Indirect call

#### **2840**

Call-with-save-area to long register

#### **2880**

Call-with-save-area to short register

#### **2891**

XLC C function call

#### **28A1**

XLC C function call in device driver

#### **28A2**

XLC C function call to SIC entry

#### **2C00**

Return-with-save-area

#### **2C10**

Indirect call return

#### **2C40**

Return-with-save-area from long register

#### **2C80**

Return-with-save-area to long register from short register

#### **2C91**

XLC C function return

#### **2CA1**

XLC C function return, error

#### **2CB1**

XLC C function return in device driver

### **3600**

Exit to the dispatcher

### **3610**

Input unpark mask

### **3611**

Switch master

### **3612**

Enter parked wait state

### **3613**

Unpark processor

### **3614**

Threading level change

### **3615**

Results of CPU Quiesce

### **3616**

Threading level change



**3617**

CPU Quiesce

**3651**

Enter enabled wait state

**FA00**

Clear plist during parsing

**ABEND**

Abend trace codes:

**FFEE**

Resume trace after soft abend

**FFFF**

Suspend trace during soft abend

**PRG**

Program interrupt entries:

**0300**

Program interrupt

**8300**

Program interrupt

**MCH**

Machine check interrupt related activities:

**0400**

Machine check interrupt

**F400**

CPU is check stopped

**F401**

Check stop CPU recovery

**I/O**

Input/output interrupt related activities:

**0501**

I/O interrupt (z/Architecture)

**1099**

I/O interrupt type 1

**10F0**

I/O sense data received

**10F1**

Concurrent sense data received

**STOR**

Host storage maintenance related activities:

**FRE**

FREE and FRET:

**0600**

Obtain free storage (FREE)

**0620**

Obtain SCSI pool storage (FREE)

**0700**

Return free storage (FRET)

**0720**

Return SCSI pool storage (FRET)

## SET CPTRACE

### **0750**

Obtain SXS page for free storage

### **0770**

Reuse existing free storage page

### **ALFR**

Host logical aligned free storage requests:

### **8620**

Obtain host logical aligned free storage

### **8720**

Return host logical aligned free storage

### **AAFR**

Absolute aligned free storage requests:

### **8630**

Obtain absolute aligned free storage

### **8730**

Return absolute aligned free storage

### **VFRE**

Virtual free storage requests:

### **0610**

Obtain virtual free storage

### **0710**

Return virtual free storage

### **4020**

64-bit host real translation

### **FRAMEREP**

Has no effect and is retained for command compatibility.

### **VMSTOR**

Virtual machine storage:

### **4010**

Virtual machine lock storage

### **4011**

Virtual machine unlock storage

### **C010**

Virtual machine lock storage (TRACG)

### **C011**

Virtual machine unlock storage (TRACG)

### **FRMFREE**

Extended free storage:

### **F704**

Return free storage backed page

### **F706**

Obtain free storage backed page

### **F708**

Return free storage frame

### **F70A**

Obtain free storage frame

### **SCH**

Adding or dropping users to or from a scheduler queue:

**3200**

Add a user to dispatch list

**3210**

Drop a user from dispatch list

**IOS**

I/O instructions:

**REALIOS**

Real I/O instructions:

**CSCH**

Clear subchannel:

**1000**

CC 0

**1003**

CC 3

**HSCH**

Halt subchannel:

**1010**

CC 0

**1011**

CC 1

**1013**

CC 3

**MSCH**

Modify subchannel:

**1020**

CC 0

**1021**

CC 1

**1023**

CC 3

**1095**

CC 0 type 1

**1096**

CC 1 type 1

**1097**

CC 2 type 1

**1098**

CC 3 type 1

**SSCH**

Start subchannel:

**1030**

CC 0

**1031**

CC 1

**1033**

CC 3

**SSCHSNS**

Start subchannel (sense):

## SET CPTRACE

**1038**

CC 0

**1039**

CC 1

**103B**

CC 3

### **TSCH**

Test subchannel:

**1050**

CC 0

**1051**

CC 1

**1053**

CC 3

**1091**

CC 0 type 1

**1092**

CC 1 type 1

**1093**

CC 3 type 1

### **RSCH**

Resume subchannel:

**1080**

CC 0

**1081**

CC 1

**1082**

CC 2

**1083**

CC 3

### **CSIO**

Channel Subsystem I/O

Reserved for IBM use:

**108F**

Extended channel information

**1090**

Channel Subsystem I/O

### **CIORF**

Cancel I/O request facility:

**10A0**

Cancel I/O

### **1C01**

Unit check

### **1C02**

Valid code for unit exception

### **1D01**

I/O related machine check

### **CHLCHEK**

Channel check:

**1D02**

Channel check

**1D03**

Channel check at termination

**1D04**

Reset channel path

**QDIO**

Queued Direct I/O instructions

Reserved for IBM use:

**1700**

QDIO SQBN state change

**1701**

QDIO invalid SQBN state detected by CP

**1702**

SIGA CC 0

**1703**

SIGA CC 1

**1704**

SIGA CC 3

**1705**

Virtual SIGA CC 0

**1706**

Virtual SIGA CC 1

**1707**

Virtual SIGA CC 3

**1708**

Real SIGA CC 2

**1709**

Virtual SIGA CC 2

**970A**

Real adapter interrupt 64

**170B**

Poll for real AIF events

**170C**

Defer AIF polling

**170D**

Post completion ID

**970D**

Post completion ID QEBSM

**170E**

AI alert

**170F**

AI alert management

**1710**

Wake-up a sleeping user

**1711**

Selected wrong user to wake

**1712**

Virtual SQBS issued

**1713**

Virtual SQBS CC 0

**1714**

Virtual SQBS CC 1

**1715**

Virtual SQBS CC 2

**1716**

Virtual SQBS CC 3

**1720**

Virtual QEBSM SIGA CC 0

**1721**

Virtual QEBSM SIGA CC 1

**1722**

Virtual QEBSM SIGA CC 2

**1723**

Virtual QEBSM SIGA CC 3

**1724**

Virtual EQBS issued

**1725**

Virtual EQBS CC 0

**1726**

Virtual EQBS CC 1

**1727**

Virtual EQBS CC 2

**1728**

Virtual EQBS CC 3

**PCI**

z/PCI I/O instructions

Reserved for IBM use:

**1200**

PCI Modify CC0 (real)

**1201**

PCI Modify CC1 (real)

**1202**

PCI Modify CC2 (real)

**1203**

PCI Modify CC3 (real)

**1204**

PCI Modify issued (virt)

**1205**

PCI Modify CC0 (virtual)

**1206**

PCI Modify CC1 (virtual)

**1207**

PCI Modify CC2 (virtual)

**1208**

PCI Modify CC3 (virtual)

**1209**

PCI Load (real)

- 120A**  
PCI Store (real)
- 120B**  
PCI Store Blk (real)
- 120C**  
PCI Event Info (real)
- 1211**  
Poll of PCI Interrupt
- 1212**  
PCI Interrupt Ctrl (real)
- 1213**  
PCI Interrupt Ctrl (virtual)
- 1220**  
Host Load PCI
- 1221**  
Host Store PCI
- 1222**  
NVMe I/O
- 1230**  
Call Logical Proc CC3 (virt)
- 1231**  
Call Logical Proc CC3 (real)
- 1232**  
Call Logical Proc (virt)
- 1233**  
Call Logical Proc (real)
- 1234**  
PCI Configure CC0 (virtual)
- 1235**  
PCI Configure CC2 (virtual)
- 1236**  
PCI Configure CC3 (virtual)
- 1237**  
PCI Configure (real)
- 123A**  
PCI Deconfig CC0 (virtual)
- 123B**  
PCI Deconfig CC2 (virtual)
- 123C**  
PCI Deconfig CC3 (virtual)
- 123D**  
PCI Deconfig (real)

**LDIOS**

Logical Device I/O instructions:

**LDCSCH**

Clear logical subchannel:

- 8000**  
CC 0

## SET CPTRACE

**8003**  
CC 3

### LDHSCH

Halt logical subchannel:

**8010**  
CC 0

**8011**  
CC 1

**8013**  
CC 3

### LDMSCH

Modify logical subchannel:

**8020**  
CC 0

**8021**  
CC 1

**8023**  
CC 3

### LDSSCH

Start logical subchannel:

**8030**  
CC 0

**8031**  
CC 1

**8033**  
CC 3

### LDSSCHS

Start logical subchannel (sense):

**8038**  
CC 0

**8039**  
CC 1

**803B**  
CC 3

### LDTSCH

Test logical subchannel:

**8050**  
CC 0

**8051**  
CC 1

**8053**  
CC 3

### 80F0

Logical I/O sense data received

### 8501

Logical I/O interrupt (z/Architecture)

### IOAIOS

I/O Assist instructions:



**IOACSCH**

I/O Assist clear subchannel:

**5000**

CC 0

**5003**

CC 3

**IOAHSCH**

I/O assist halt subchannel:

**5010**

CC 0

**5011**

CC 1

**5012**

CC 2

**5013**

CC 3

**IOAMSCH**

I/O assist modify subchannel:

**5020**

CC 0

**5021**

CC 1

**5022**

CC 2

**5023**

CC 3

**IOASSCH**

I/O assist start subchannel:

**5030**

CC 0

**5031**

CC 1

**5032**

CC 2

**5033**

CC 3

**IOATSCH**

I/O assist test subchannel:

**5050**

CC 0

**5051**

CC 1

**5053**

CC 3

**IOATPI**

I/O assist test pending interrupt:

**5060**

CC 0

**5061**

CC 1

**IOARSCH**

I/O assist resume subchannel:

**5080**

CC 0

**5081**

CC 1

**5082**

CC 2

**5083**

CC 3

**IOACIORF**

I/O Assist Cancel I/O Request Facility:

**50A0**

I/O assist Cancel I/O

**5501**

I/O assist I/O interrupt (z/Architecture)

**IOR**

Stacking or unstacking of IORBKs or TRQBKs:

**3000**

Stack IORBK/TRQBK

**3010**

Unstack IORBK/TRQBK

**CPX**

Stacking or unstacking of CPEBKs and related activity:

**3300**

Stack CPEBK

**3310**

Unstack CPEBK

**3700**

Stack work bits

**RUNU**

Virtual machine dispatching activity:

**8A00**

Run user

**8B00**

Reflect a guest program interruption via HCPGIRPG

**8B01**

Reflect a guest program interruption via HCPGIRPC

**8B02**

Reflect a guest program interruption via HCPGIRPI

**VSIE**

Virtual SIE activities:

**9A00**

Run user in virtual SIE (z/Architecture)

**9A01**

Reflect a Vguest program interruption via HCPWPR

**9A11**

Virtual SIE interception (z/Architecture)

**B500**

Interception, not instruction (z/Architecture)

**B504**

Interception, instruction (z/Architecture)

**VINT**

Simulated I/O interrupts:

**0C01**

Virtual I/O interrupt (z/Architecture)

**0C02**

Virtual adapter interruption

**0D00**

Virtual CSW stored

**2500**

Guest I/O untranslation

**VCSW**

Simulated I/O instructions:

**0C32**

Virtual modify subchannel

**0C33**

Virtual start subchannel

**0C35**

Virtual test subchannel

**0C36**

Virtual test pending interrupt

**0D90**

Virtual start I/O

**0D91**

Virtual start I/O fast

**APPCIUCV**

APPC/VM and IUCV activities:

**IUCV**

IUCV activities:

**1500**

IUCV Query function

**1501**

IUCV Test message function

**1502**

IUCV Retrieve buffer function

**1503**

IUCV Describe function

**1504**

IUCV Send function

**1505**

IUCV Receive function

**1506**

IUCV Reply function

## SET CPTRACE

### **1507**

IUCV Test completion function

### **1508**

IUCV Reject function

### **1509**

IUCV Purge function

### **150A**

IUCV Accept function

### **150B**

IUCV Connect function

### **150C**

IUCV Declare buffer function

### **150D**

IUCV Quiesce function

### **150E**

IUCV Resume function

### **150F**

IUCV Sever function

### **1510**

IUCV Set mask function

### **1511**

IUCV Set control mask function

### **1515**

IUCV Interrupt poll function

### **1530**

Have an IUCV Interrupt

### **1531**

IUFQA driven for a System Service

### **1550**

IUMRF driven but interrupt not found or User not enabled for interrupts

## **APPC**

APPC/VM activities:

### **1404**

APPCVM SENDxxxx function

### **1405**

APPCVM RECEIVE function

### **140B**

APPCVM CONNECT function

### **140F**

APPCVM SEVER function

### **1412**

APPCVM QUERY STATE function

### **1413**

APPCVM SET MODIFY function

### **1414**

APPCVM SET STATE function

### **142B**

APPCVM CONNECT Resume Suspended

**1430**

Have an APPCVM Interrupt

**143B**

APPCVM CONNECT Resume Unnecessary

**CCS**

SNA/CCS activities:

**CCSN**

CCS normal events:

**1600**

CCS ACCEPT

**1603**

CCS PURGE

**1604**

CCS RECEIVE

**1606**

CCS REPLY

**1608**

CCS SEND 1-way

**1609**

CCS SEND 2-way

**160A**

CCS SEVER

**160B**

CCS Logic error in CCS interface CB

**160C**

CCS REPLY from VSM

**160E**

CCS CONNECT from VSM

**1610**

CCS SEVER from VSM

**1611**

CCS Message complete from VSM, 1-way SEND

**1612**

CCS Connect from VSM for LU

**1613**

CCS Logic error in VSM interface CB

**1614**

CCS Error in user environment, SEVER user

**1615**

CCS SNA control block chain not valid

**CCSE**

CCS error events:

**1680**

CCS ACCEPT error

**1684**

CCS RECEIVE error

**1686**

CCS REPLY error

**1688**

CCS SEND 1-Way error

**1689**

CCS SEND 2-Way error

**ISFC**

ISFC activities:

**7000**

ISFC Check APPC/IUCV path

**7002**

ISFC Locate PTHBK by APPC path

**7003**

ISFC Locate PTHBK by ISFC session

**7004**

ISFC Add message (MDEBK) to work queue

**7005**

ISFC Get message (MDEBK) from a work queue

**7008**

ISFC Queue a message for a link

**7009**

ISFC Dequeue a message for a link

**700A**

ISFC Check CP Application path

**700B**

ISFC Add message (CARBK) to work queue

**700C**

ISFC Get message (CARBK) from a work queue

**700D**

ISFC Node synchronization start

**700E**

ISFC Node synchronization complete

**700F**

ISFC Connectivity change notify

**7010**

ISFC Send message failed

**7011**

ISFC Socket created

**7012**

ISFC Socket destroyed

**7013**

ISFC Remove a logical link

**7014**

ISFC Remove a link device

**7015**

ISFC Message timed out

**7017**

ISFC \*SYNC Send message failed

**7018**

ISFC Status change

- 7019**  
ISFC Delay obtaining storage
- 701A**  
ISFC Connect device to link
- 701B**  
ISFC Disconnect device from link
- 7020**  
ISFC Unmatched Input Message
- 7021**  
ISFC Unexpected I/O Status
- 7022**  
ISFC LWKBK sequence number error
- 7023**  
ISFC Data format error
- 7024**  
ISFC No matching end point found for message
- 7025**  
ISFC Link load sampling
- 7026**  
ISFC Late response notice
- 7027**  
ISFC Out of storage
- 7028**  
Reserved for IBM use.
- 702A**  
ISFC Inbound frame error
- 702B**  
ISFC Device error event
- 702C**  
ISFC Duplicate node ID
- 7030**  
ISFC Transport mode change
- 7031**  
ISFC Node loss cleanup

**VNET**

Virtual Networking activities:

- 7400**  
\*VSWITCH Controller Message
- 7401**  
MAC Address Management
- 7402**  
Virtual Switch Bridge Port Outbound Transfer
- 7403**  
Virtual Switch Bridge Port Completion Event
- 7404**  
Virtual Switch Bridge Port Inbound Transfer
- 7405**  
Shared Port Group Parameter change

## SET CPTRACE

### **7406**

Shared Port Group Take-Over change

### **7407**

IVL Membership change

## **VCTC**

Virtual CTCA activities:

### **7900**

VCTC define

### **7901**

VCTC detach

### **7902**

VCTC couple

### **7903**

VCTC interface disconnect

### **7904**

VCTC selective reset

### **7905**

VCTC system reset

### **7906**

VCTC virtual I/O simulation

### **7907**

VCTC unsolicited interrupt

## **DUMP**

SNAPDUMP processing activities:

### **F800**

Snapdump invoked via the command

### **F801**

Snapdump invoked via HCPABEND macro

### **F802**

Snapdump completed

## **EXIT**

CP Exit activities:

### **F900**

Call exit start

### **F910**

Call exit routine start

### **F920**

Call exit routine finish

### **F930**

Call exit finish

### **F940**

Call exit routine (nop)

## **MDC**

Minidisk cache activities:

### **4100**

MDC page default

### **4101**

MDC cache miss exit CC=0



**4102**

MDC cache miss exit CC=3

**4103**

MDC cache miss I/O required

**4104**

MDC cache miss I/O complete CC=0

**4105**

MDC deferring CPEBK

**C106**

MDC main storage steal complete

**4108**

MDC I/O retry as nonstandard

**4109**

Reserved for IBM use.

**CSIM**

MDC control unit simulation activities:

**4110**

MDC cache simulation entry

**4111**

MDC cache simulation abort

**4112**

MDC cache simulation successful exit

**4113**

Reserved for IBM use.

**4114**

Reserved for IBM use.

**4115**

MDC track purge for Diag/Block I/O

**4116**

MDC results from Diag/Block I/O

**4117**

Reserved for IBM use.

**4118**

Reserved for IBM use.

**4119**

Reserved for IBM use.

**CONTAINR**

Container platform specific activities:

**6000**

Lock

**6001**

Lock

**6002**

Unlock

**6003**

Unlock

**6004**

Test lock

## SET CPTRACE

### **6005**

Allocate lock

### **6006**

Deallocate lock

### **6010**

Latch entry

### **6011**

Latch exit

### **6012**

Unlatch entry

### **6013**

Unlatch exit

### **6014**

Test latch

### **6015**

Allocate latch

### **6016**

Deallocate latch

### **6020**

Wait event entry

### **6021**

Wait event exit

### **6022**

Notify event entry

### **6023**

Notify event exit

### **6024**

Allocate event

### **6025**

Deallocate event

### **6026**

SCSI response

### **6027**

SCSI command

### **6028**

SCSI I/O data

### **6029**

SCSI support command

### **6030**

SCSI end

## **SERVICE**

VM Service diagnostic trace entries (usually off):

### **F000**

VM Service Trace Entry

### **F001**

VM Service Trace Entry

### **F002**

VM Service Trace Entry

**F003**  
VM Service Trace Entry

**F004**  
VM Service Trace Entry

**F005**  
VM Service Trace Entry

**F006**  
VM Service Trace Entry

**F007**  
VM Service Trace Entry

**F008**  
VM Service Trace Entry

**F009**  
VM Service Trace Entry

#### **VTIMING**

Message activities

Reserved for IBM use:

**C019**  
Send message

**C01A**  
Test message state

**C01B**  
Modify MSG subchannel

**C01C**  
Store MSG subchannel

**C01D**  
Test MSG subchannel

**C01E**  
Clear MSG subchannel

**C043**  
Prepare channel buffer

**C044**  
Signal channel buffer

**C045**  
Move channel buffer data

**C046**  
Test channel buffer

**C047**  
Locate channel buffer

**C048**  
Move channel BUF multi

**C049**  
Failed making external-call-IRPT pending on CFCC userid

#### **SPINLOCK**

Formal spin lock trace codes (see usage note [“5” on page 1470](#)):

**3400**  
Get a spin lock in exclusive mode

**3401**  
Get a spin lock in exclusive mode via HCPSYNAX

## SET CPTRACE

### 3402

Release a spin lock from exclusive mode

### 3403

Get a share of a spin lock

### 3404

Get a spin lock in shared mode via HCPSYNAS

### 3405

Release a share of a spin lock

### 3406

Change an exclusive spin lock to shared mode

### 3407

Change a shared spin lock to exclusive mode

## SSI

Cluster Operations:

### 1901

Evaluate synchronization request

### 1902

Connectivity change

### 1903

Synchronization SSI function

### 1904

Receive response

### 1905

Receive synchronization request

### 1906

Local state change

### 1907

Reply to synchronization request

### 1908

Synchronize socket function

### 1909

Synchronization return code

### 1910

Join event results

### 1911

Service call results

### 1912

Remote system state change

### 1913

Received a suspend notification

### 1914

Called to suspend local system

### 1915

Changed State in PDR

## USR

USERID Coordination Trace Entries:

### 1950

Userid socket send broadcast occurred

**1951**

Userid socket receive work

**1952**

Userid relocation call received

**1953**

Userid state change received

**1954**

Userid delete hash table entry or storage occurred

**1955**

Userid join failure occurred

**1956**

Userid broadcast retry occurred

**1957**

Userid socket reply error occurred

**1958**

Userid join wrap-up received

**VAA**

Virtual architecture trace codes:

**2000**

Acquire VAAXLK lock locally

**2001**

Acquire VAAXLK lock globally

**2002**

Release VAAXLK lock

**2003**

Relocation domain synchronization received an SSI state change notification

**2004**

HCPXLKUU successful call

**2005**

HCPXLKUU failed call

**LGRMM**

Live Guest Relocation (LGR) Memory Move Trace Codes:

**2100**

End-to-end throttle: Activation message received (LGR source)

**2101**

End-to-end throttle: Deactivation message received (LGR source)

**2102**

End-to-end throttle: Memory move has resumed (LGR source)

**2103**

Logical link throttle: "Pages on aux" processor has resumed (LGR source)

**2104**

Logical link throttle: PGMBK walker has resumed (LGR source)

**2105**

Logical link throttle: DAT walk has resumed (LGR source)

**2106**

"Pages on aux" processor has finished, allowing DAT walk to resume (LGR source)

**2107**

Paging overrun resolved; PGMBK walker resumed (LGR source)

**RESET**

ends specific tracing for the indicated virtual machine, system work, or the entire SPECIFIC group. The virtual machine or machines (and/or SYSTEM) become members of the NONSPECIFIC group. SET CPTRACE FOR SPECIFIC RESET, therefore, ensures that all virtual machines and SYSTEM are in the NONSPECIFIC group.

**Usage Notes**

1. You may combine several commands into one command as long as the FOR qualifier is the same. For example, `set cptrace for nonspecific on svc off ext`.

**Note:** FOR NONSPECIFIC is the default; the following command has the same result: `set cptrace on svc off ext`.

2. If you start a SPECIFIC trace, you must be aware of the initial state of the trace, or you get whatever the initial state was, modified by the NONSPECIFIC trace. To find out the initial trace state, enter `query cptrace for specific`; then, check the response and enter `query cptrace`.

(QUERY CPTRACE gives you the NONSPECIFIC state.)

3. You cannot disable tracing for entries that can occur at the time of a CP trace table full condition. The two entry conditions that can occur are "PROGRAM INTERRUPTION" (0300) for trace table page full, program interrupt X'16', and "TRACE TABLE PAGE FULL DURING MACHINE CHECK HANDLING" (F40F). The TIME STAMP TRACE ENTRY (FFFD) cannot be disabled by the SET CPTRACE command either.
4. The SET CPTRACE command processes categories sequentially from left to right; tracing for the first category begins before the second category is processed. Therefore, it may be possible to have trace settings in effect during command processing that are different from the trace settings before and after command processing.

The NONSPECIFIC trace setting, modified by the command initially creating the SPECIFIC trace, is the initial state of a SPECIFIC trace setting.

To get an initial state of ALLCODES enabled, enter: `set cptrace for userid on allcodes`, combined with, or followed by, the desired trace settings.

5. SPINLOCK trace entries are generated only for spin locks whose obtain and release macros indicate they should be traced.

**Examples**

To disable the trace for IUCV instructions, enter: `set cptrace off iucv`. As an example, assume that when your system is IPLed, all users are in the NONSPECIFIC category. If you want to put all users in the NONSPECIFIC category and you want to trace all trace codes, enter: `set cptrace for nonspecific on allcodes`. If you enter the command with all its defaults and truncations specified, the command looks like this: `set cptr`. If you want to put some users in the SPECIFIC category, enter the following commands: `set cptrace for nonspecific off all`. This turns off all trace codes for users in the NONSPECIFIC category.

Perhaps you want to set a SPECIFIC trace for users CMS000 and SYSTEM. For CMS000 you want to set ON code categories I/O and IUCV, and for SYSTEM you want to set ON only the code categories SVC and IUCV. (See the trace code category listings in [“category” on page 1447](#).)

To do this, enter the following commands:

```
set cptrace for cms000 off on i/o
set cptrace for system off on svc
set cptrace for specific on iucv
```

If you now want to put the user CMS000 back into the NONSPECIFIC group, enter: `set cptrace for cms000 reset`. The system traces SYSTEM for SVC and IUCV trace code categories.

**Note:** You can only use RESET to set users or SYSTEM back into the NONSPECIFIC group. In order to set codes for a NONSPECIFIC user off, enter `set cptrace for userid off code/category`.

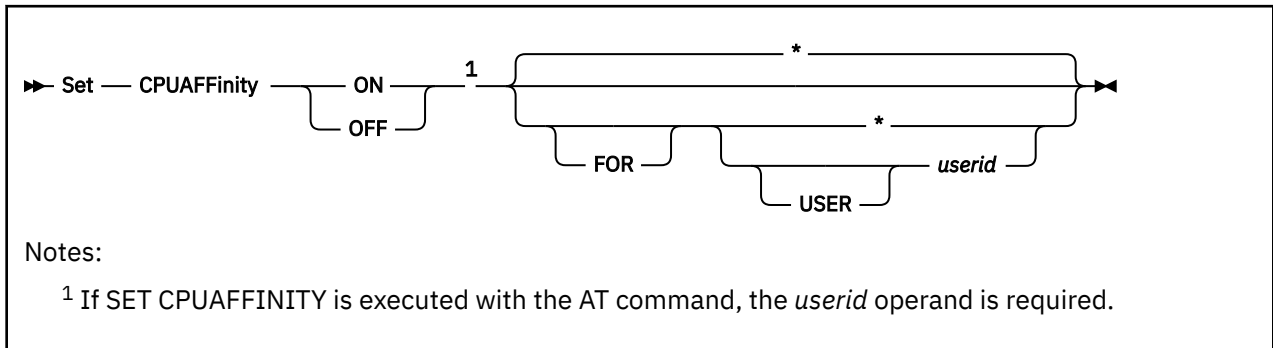
If you want to trace ALLCODES for one SPECIFIC user (a user ID) only, enter the following commands:

```
set cptrace for all off allcodes  
set cptrace for userid on allcodes
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP053E [XAUTOLOG failed for *userid*:] {*userid*/*value*} not in CP directory
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1033E No SPECIFIC CP TRACE is in effect.
- HCP1034I No SPECIFIC CP TRACE is in effect for *userid*.

## SET CPUAFFINITY



### Authorization

Privilege Class: A

### Purpose

Use SET CPUAFFINITY to change the CPU affinity setting for a user.

### Operands

#### USER *userid*

identifies the virtual machine whose CPU affinity you are setting.

\*

specifies that CPU affinity is to be set for the *userid* issuing the command.

#### ON

#### OFF

specifies if CPU affinity should be turned ON or OFF. Initially, CPU affinity is set to ON for users at logon time. ON means the virtual CPUs are dispatched on only real CPUs of matching type. When CPU affinity is OFF, the guest's virtual processors of secondary type always get dispatched on host real processors of primary type.

### Usage Notes

1. CPU affinity is a global setting for the entire virtual CPU configuration. CPU affinity is achieved when there is a real CPU with the same type as the virtual secondary processor trying to obtain affinity. If no real CPU of the same type exists, then CPU affinity is suppressed for that virtual processor type.
2. The resource pooling function requires all users in a resource pool to use the same CPU type that is being limited by the resource pool. By requiring CPU affinity to be on for a user ID to be assigned to a resource pool, z/VM handles all users in the resource pool in an identical manner in cases where CPU affinity needs to toggle between on and suppressed, and ensures this change occurs for all affected users at the same time. After a user is assigned to a resource pool, CPU affinity cannot be set off for that user.

### Responses

Response 1:

```
CPU Affinity is suppressed for CPU type type
```

There was no real processor with the same type as the virtual secondary processor.



**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option* | command contains extra option(s) starting with *option*}
- HCP045E *userid* not logged on
- HCP1014E A required option is missing.
- HCP1463I CPU Affinity is suppressed for CPU type *type*.
- HCP1473E CPU Affinity cannot be turned OFF for a user in a resource pool
- HCP5780E Adjunct does not support the *command subcommand* command

## SET CPUID

---

►► Set — CPUid — *number* ◄◄

### Authorization

Privilege Class: G

### Purpose

Use SET CPUID to change the processor identifier for the virtual processor from which the command is issued.

### Operands

#### *number*

changes the virtual processor identifier. The 6 hexadecimal digits you specify in *number* become the new processor identifier number or processor serial number.

1. The SET CPUID command changes bits 8 through 31 of the CPU identifier that is set by the STIDP instruction to the value you specify.
2. If the guest owning the virtual processor is relocated, the serial number of the virtual processor does not change as a result of the relocation, even if the FORCE ARCHITECTURE or FORCE DOMAIN options were used on the VMRELOCATE command.

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP263E Too many operands were supplied for this command.
- HCP1005E An invalid character was encountered.

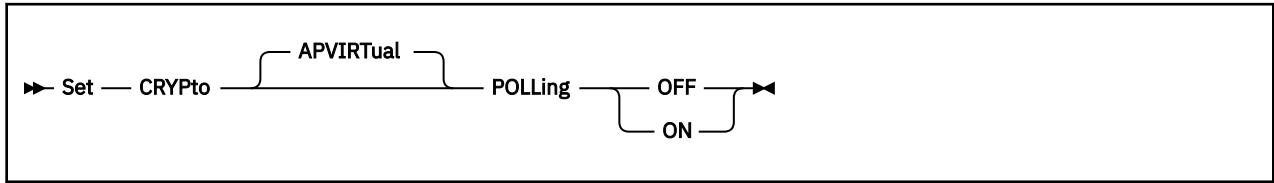
## SET CPUPOOL

---

### **Purpose**

SET CPUPOOL has been superseded by SET RESPOOL. See [“SET RESPOOL”](#) on page 1679.

## SET CRYPTO



### Authorization

Privilege Class: B

### Purpose

Use SET CRYPTO APVIRTUAL POLLING to specify whether CP uses polling or interruptions to drive the processing of crypto work that is assigned to the shared pool.

### Operands

#### POLLing ON

CP uses polling to drive the processing of APVIRT crypto work.

#### POLLing OFF

CP uses crypto interruptions to drive the processing of APVIRT crypto work. CP does not use polling to drive the processing of APVIRT crypto work.

### Usage Notes

1. The command is allowed even if no shared pool is defined. The setting becomes active as soon as a shared resource is attached to the system.
2. The command affects CP's processing of work for APVIRT crypto users. The command does not affect the way that a virtual machine places work in the shared pool or receives replies from crypto work that is processed by shared crypto resources.
3. Use the QUERY CRYPTO POLLING command to determine the current setting. See [QUERY CRYPTO](#) in *z/VM: CP Commands and Utilities Reference*.
4. Use the CRYPTO APVIRTUAL POLLING statement in the system configuration file to specify the setting that takes effect at system IPL time. See [CRYPTO APVIRTUAL Statement](#) in *z/VM: CP Planning and Administration*.
5. If a POLLING setting is not specified on a CRYPTO APVIRTUAL statement in the system configuration file, then the default setting is OFF.

### Examples

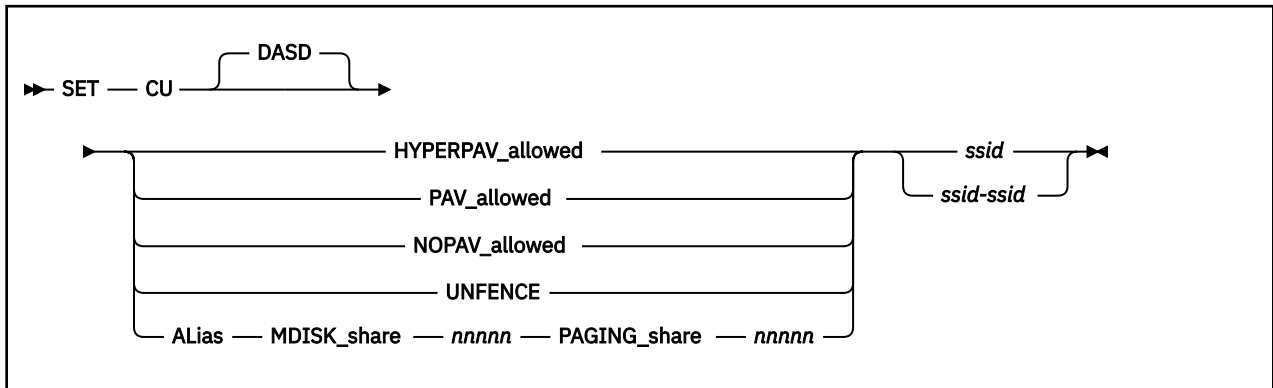
- SET CRYPTO APVIRTUAL POLLING ON
- SET CRYPTO POLLING OFF

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra options starting with *option*
- HCP1014E A required option is missing
- HCP1713E A required facility is not available to allow Crypto to operate with polling off. Polling remains on.

- HCP1728E AP Crypto resources are not installed on this system.

## SET CU



### Authorization

Privilege Class: B

### Purpose

Use SET CU to change the functional level at which a specified control unit is allowed to operate with respect to your z/VM system (the system on which you issue the command). For the parameters you specify, you can permit the control unit to operate below, at, or (for PAV\_allowed) above its currently-installed functional level. You can allow the control unit to reset soft fence state for the DASDs on it. You can also specify the relative share of system-attached HyperPAV aliases to be used for minidisk I/O and paging I/O.

### Notes:

- The command applies only to first-level z/VM systems. When z/VM is running as a guest of another z/VM system, the guest z/VM system does not control the operation of the control unit. If the command is issued to a guest z/VM system, the command is rejected.
- The control unit that is associated with an EDEVICE HyperPAV pool has a subsystem identifier (SSID) that cannot be referenced by the SET CU command and thus cannot be affected by the SET CU command.

### Operands

#### HYPERPAV\_allowed

tells CP to allow a specified control unit to operate with HyperParallel Access Volume (HyperPAV) devices in your z/VM system.

**Note:** All alias devices in a specified control unit must be offline to CP if changing to the HYPERPAV\_allowed setting.

#### PAV\_allowed

tells CP to allow a specified control unit to operate with Parallel Access Volume (PAV) devices in your z/VM system.

**Note:** All alias devices in a specified control unit must be offline to CP if changing from the HYPERPAV\_allowed setting.

#### NOPAV\_allowed

tells CP to not allow a specified control unit to operate with PAV or HyperPAV devices with respect to your z/VM system.

**Note:** All alias devices in a specified control unit must be offline to CP if changing to the NOPAV\_allowed setting. After the command, all alias devices will become non-operational.

**UNFENCE**

tells CP to allow a specified control unit to reset soft fence state for all the devices on it.

**Alias**

tells CP the command applies to aliases that are associated with the control unit. When this operand is specified, MDISK\_share and PAGING\_share are also required in the order shown in the command syntax diagram.

**MDISK\_share *nnnnn***

specifies the relative share that minidisk I/O will use when it competes for system-attached HyperPAV aliases in the control unit. The fraction of the total number of system-attached HyperPAV aliases that CP will attempt to make available for use by minidisk I/O is *nnnnn* divided by the sum of the values of the MDISK\_share and PAGING\_share operands. The value of *nnnnn* ranges from 0 to 10000. When this operand is not specified, the value of MDISK\_share is 0. When the value of MDISK\_share is 0, minidisk I/O is not promised any share of the control unit's system-attached HyperPAV aliases.

**PAGING\_share *nnnnn***

specifies the relative share that paging I/O will use when it competes for system-attached HyperPAV aliases in the control unit. The fraction of the total number of system-attached HyperPAV aliases that CP will attempt to make available for use by paging I/O is *nnnnn* divided by the sum of the values of the MDISK\_share and PAGING\_share operands. The value of *nnnnn* ranges from 0 to 10000. When this operand is not specified, the value of PAGING\_share is 0. When the value of PAGING\_share is 0, paging I/O is not promised any share of the control unit's system-attached HyperPAV aliases.

***ssid******ssid-ssid***

is the subsystem identifier (SSID) of a control unit as established in a control unit during its installation. The SET CU command operates on the control unit that is identified. The variable *ssid* must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single subsystem identifier, a list, a range, or any combination thereof.

**Usage Notes**

1. The actual capabilities of control units that support the PAV or HyperPAV functions (such as the IBM DS6000™ and DS8000® series) are defined by the presence or absence of the applicable features in these control units. CP's default value for a control unit's allowed operational level is PAV\_allowed. However, if a control unit can function at the HYPERPAV level, then the default value is HYPERPAV\_allowed. The allowed level can be changed by CU statements in your system configuration file or by SET CU commands you issue. For example, if a control unit has the HyperPAV feature installed but a CU statement in the system configuration file specifies PAV\_allowed, then any alias devices defined in the control unit will operate as PAV devices and not as HyperPAV devices.
2. The allowed level cannot be higher than the installed level, except that the allowed level can be PAV\_allowed when neither the PAV nor the HYPERPAV features are installed in a control unit. In this latter case, if you later install the PAV feature in a control unit while it continues in normal operation, any defined alias devices will become available to CP without the need for a SET CU command. If you additionally install the HYPERPAV feature, a SET CU command is necessary to raise the allowed functional level to HYPERPAV\_allowed.
3. If an allowed level is lower than the installed level, the control unit will function (with respect to your z/VM system) at the allowed level only.
4. If the HYPERPAV feature is removed from a control unit while HYPERPAV alias devices are being used, they become unusable and the control unit notifies CP of the change. The error recovery for these devices is to take them offline to CP and to lower the allowed functional level for the control unit to PAV\_allowed.
5. If the PAV feature is removed from a control unit while PAV alias devices are being used, they become non-operational. If the PAV feature is subsequently restored, the alias devices become operational again as long as you did not change the PAV\_allowed setting.
6. When SET CU is used to change the PAV state of a control unit and the applicable features are already installed, the following occurs:

- *NOPAV-to-PAV transition* - previously-defined alias devices can be brought online and become associated with their applicable base pools.
  - *NOPAV-to-HyperPAV transition* - previously-defined alias devices can be brought online and become associated with their applicable HyperPAV pools.
  - *PAV-to-HyperPAV transition* - affected alias devices are disassociated from their base devices; when brought online, they become associated with their applicable HyperPAV pools.
  - *HyperPAV-to-PAV transition* - affected alias devices are disassociated from their HyperPAV pools; when brought online, they become associated with their applicable base devices.
  - *HyperPAV-to-NOPAV transition* - affected alias devices are disassociated from their HyperPAV pools and taken offline; they cannot be VARYed ONline.
  - *PAV-to-NOPAV transition* - affected alias devices are disassociated from their base pools and taken offline; they cannot be VARYed ONline.
7. The UNFENCE operand will reset the soft fence state of every configured DASD on the control unit regardless of whether or not they are defined to the VM LPAR.
  8. The MDISK\_share and PAGING\_share operands influence CP in choosing an I/O operation to be executed on a system-attached HyperPAV alias device. The settings of these operands have meaning to CP only when all system-attached HyperPAV aliases associated with the control unit are currently busy executing I/O and an alias becomes available.

The default behavior for CP when an alias becomes available is to scan for I/O queued on the HyperPAV base devices on the control unit. The first I/O operation found that can be executed on a system-attached HyperPAV alias device is chosen for execution (this is the behavior when the values of MDISK\_share and PAGING\_share are 0). By specifying a non-zero value for MDISK\_share, PAGING\_share, or both, you specify an entitlement for the respective use. CP makes sure an entitlement is met before resorting to its default behavior.

The entitlement computation rounds the value to the nearest whole number. Specifically:

- A value with a decimal component of less than .5 is rounded down.
- A value with a decimal component of .5 or greater is rounded up.

However, if the sum of the resulting entitlements for minidisk I/O and paging I/O exceeds the number of system-attached HyperPAV aliases, both entitlements are rounded down and the default algorithm applies to the residual alias.

To determine the alias activity for a HyperPAV base device, examine the device monitor data. To see minidisk I/O's and paging I/O's current shares and entitlements for a control unit, use the QUERY CU command with the ALIAS\_Share operand. To obtain a complete list of alias devices that are associated with a control unit, use the QUERY CU command with the ALIASES operand.

For ECKD CUs, some performance specialists suggest not mixing paging I/O and minidisk I/O in the same CU, even though SET CU ALIAS mediates alias use correctly. Results will vary from one workload to another. If the results that you get when mixing do not meet expectations, try placing paging I/O and minidisk I/O in separate LCUs.

For EDEVICE CUs, the SET CU ALIAS command does not apply, so it is possible in theory for one type of I/O to starve another type's exploitation of aliases. For such CUs, IBM recommends not mixing I/O types when I/O rates are high enough to exhaust the alias pool routinely. To watch the alias pool, use Performance Toolkit FCX327 HPALIAS or equivalent.

9. The z/VM system programmer and the DASD management programmer must work together to make sure each logical subsystem (LSS) that is being used on the system has a unique SSID.

## Examples

Example 1:

To allow the operation of HyperPAV devices on a control unit, enter the following:



```
set cu hyperpav 4343
Command complete
Ready;
```

**Example 2:**

To disallow the operation of PAV or HyperPAV devices on a range of control units, enter the following:

```
set cu nopav 4400-4410
Command complete
Ready;
```

**Example 3:**

To reset soft fence state for the devices on a control unit, enter the following:

```
set cu unfence 0409
Command complete
Ready;
```

**Example 4:**

To set a minidisk alias share of 100 and a paging alias share of 200 for control unit 4343, specify the following:

```
set cu alias mdisk_share 100 paging_share 200 4343
Command complete
Ready;
```

In this example, if the number of HyperPAV system-attached aliases is 17, the minidisk entitlement is 6 and the paging entitlement is 11.

The minidisk entitlement calculation is:

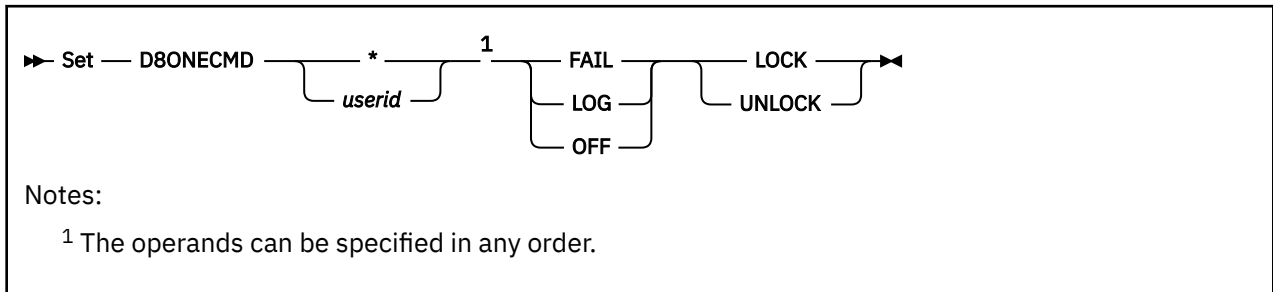
$$(100 / (100 + 200)) * 17 = 17 / 3 = 5.67 \text{ (rounded up to 6)}$$

The paging entitlement calculation is:

$$(200 / (100 + 200)) * 17 = 34 / 3 = 11.33 \text{ (rounded down to 11)}$$
**Messages**

- HCP002E Invalid operand - *operand*
- HCP6004E This command is not supported on your system.
- HCP6706E Invalid address range - *range*
- HCP6869E DASD CU *ssid* alias *rdev* is attached to a guest
- HCP6869E DASD CU *ssid* alias *rdev* is attached to the system
- HCP6869E DASD CU *ssid* alias *rdev* is not offline
- HCP6869E DASD CU *ssid* HyperPAV mode not allowed because the associated channel paths are not all of type FICON
- HCP6869E DASD CU *ssid* HyperPAV mode not supported
- HCP6869E DASD CU *ssid* no online base devices found
- HCP6869E DASD CU *ssid* PAV functions do not apply
- HCP6869E DASD CU *ssid* unable to establish new PAV mode
- HCP6869E DASD CU *ssid* does not support soft fence.
- HCP6869E DASD CU *ssid* failed to reset soft fence.

## SET D8ONECMD



### Authorization

Privilege Class: A, G

### Purpose

Use SET D8ONECMD to change the D8ONECMD settings for your own virtual machine or another user's virtual machine. D8ONECMD settings determine whether CP accepts multiple CP commands imbedded in a single CP command issued through DIAGNOSE code X'08'.

When a user uses a general VMCF-capable program to send a request that contains imbedded X'15' characters to a service user ID, the service user ID assumes the request was issued with the CP SMSG facility. The CP SMSG facility is trusted because CP ensures the service user ID does not receive any imbedded X'15' characters. When the service user ID uses DIAGNOSE code X'08' to issue a CP command that contains the request, CP interprets the X'15' characters as command separators and breaks what the service user ID thought was a single command into several commands.

SET D8ONECMD lets you to prevent the execution of "Trojan Horse" commands (commands hiding inside legitimate requests) and records any attempt to process such commands.

For information on how to display the D8ONECMD settings of your virtual machine or another user's virtual machine, see ["QUERY D8ONECMD"](#) on page 865.

### Operands

#### *userid*

indicates that you wish to change another virtual machine's D8ONECMD settings. If you are a class G user, you can specify only your own user ID.

\*

indicates that you wish to change your own virtual machine's D8ONECMD settings.

#### **FAIL**

tells CP to reject any attempt to process a DIAGNOSE code X'08' containing multiple CP commands and send a message to operations personnel. If you specify FAIL, you cannot specify LOG or OFF.

#### **LOG**

tells CP to send a message to operations personnel every time the specified user tries to process a DIAGNOSE code X'08' containing multiple CP commands. If you specify LOG, you cannot specify FAIL or OFF.

#### **OFF**

tells CP the specified user can process a DIAGNOSE code X'08' containing multiple CP commands with no restrictions. If you specify OFF, you cannot specify FAIL or LOG.

#### **LOCK**

tells CP to only let a privileged user, who is also not the command issuer, change the settings. If you specify LOCK, you cannot specify UNLOCK.

**UNLOCK**

tells CP to let a privileged user change the command filtering level for the selected virtual machine (even if the target virtual machine is the command issuer's virtual machine). If you specify UNLOCK, you cannot specify LOCK.

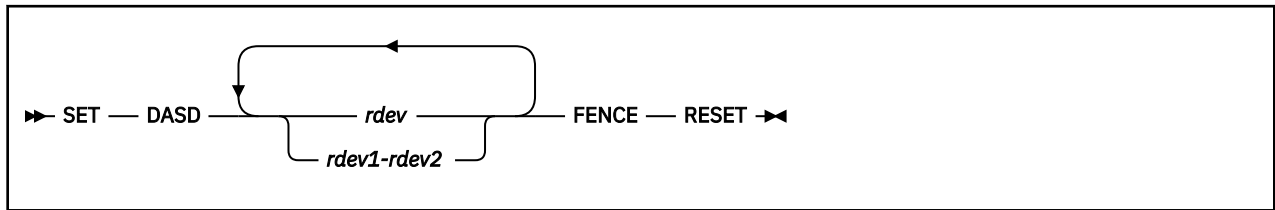
**Usage Notes**

1. The initial D8ONECMD settings for your virtual machine are determined by options set in your user directory entry.
2. This command does not affect the user's ability to enter multiple commands by typing them at the console; it affects only multiple commands issued by programs.
3. This facility does not affect a user's ability to use a Program Function Key to enter more than one command at once, nor does it keep a program from establishing a value for a Program Function Key that issues multiple commands when the user at the terminal presses the Program Function Key.
4. If you are running an earlier level of CMS and D8ONECMD is set to FAIL:
  - some CMS commands (such as SENDFILE, NETDATA, DISK LOAD, and READCARD) may not work properly
  - an earlier level of the programmable operator facility will not initialize.

**Messages**

- HCP003E Invalid option - *option*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP6889E *userid*'s DIAGNOSE X'08' options are locked.

## SET DASD



### Authorization

Privilege Class: B

### Purpose

Use SET DASD to reset soft fence state for one or more real devices.

### Operands

*rdev*

*rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers. The variable *rdev* must be a hexadecimal number between X'0000' and X'FFFF'.

### FENCE RESET

tells CP to reset soft fence state for the specified device or devices.

### Usage Notes

1. The FENCE RESET function is allowed only for dedicated DASD and fullpack minidisks. If this command is issued from a second-level system to reset soft fence state for the specified device, it will work only when the underlying first-level disk is a dedicated DASD or a fullpack minidisk. Additionally, for a guest to issue this function, the guest must have OPTION DEVMAINT or MAINTCCW user authority.

### Examples

Example 1:

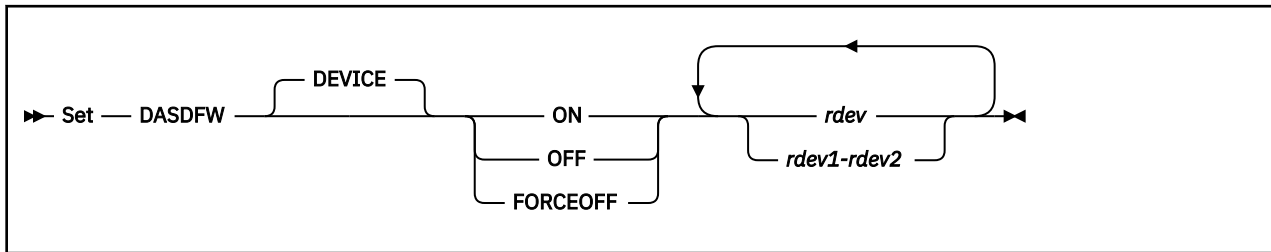
To reset a soft fence state for devices 8001-8004, enter the following:

```
set dasd 8000-8004 fence reset
Command complete;
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP1172E No action for device *rdev* because control unit does not support soft fence
- HCP1173E Reset soft fence operation failed for device *rdev*
- HCP1173E Reset soft fence operation failed for devices: *rdev-rdev*

## SET DASDFW



### Authorization

Privilege Class: B

### Purpose

Use SET DASDFW to activate or deactivate the DASD fast write function for devices in a DASD subsystem. This command is valid for cache storage subsystems that support the DASD fast write function.

**Note:** Although the SET DASDFW ON command activates the DASD fast write function, it does not necessarily cause immediate DASD fast write operation. For the DASD fast write function to operate, both the subsystem cache and nonvolatile storage must be available and both caching and DASD fast write must be activated for the device.

### Operands

#### DEVICE

specifies that the DASD fast write function is to be activated or deactivated for the specified devices. It is the default (and only) value.

#### ON

indicates that the DASD fast write function is to be activated for the specified devices.

#### OFF

indicates that the DASD fast write function is to be deactivated for the specified devices.

#### FORCEOFF


indicates that the DASD fast write function is to be unconditionally deactivated for the specified device, regardless of any internal subsystem errors. This operand should only be used when absolutely necessary, because there is a possibility of loss of data.

#### *rdev*

#### *rdev1-rdev2*

is the real device number, a list of real device numbers, or a range of real device numbers of devices for which the DASD fast write function is to be activated or deactivated.

### Usage Notes

1. The SET DASDFW command is only valid for storage subsystems that contain nonvolatile storage (NVS) and support the DASD fast write function. An error message is generated if the subsystem containing a specified device does not support this function.
2. If the NVS or cache is not available when the SET DASDFW ON command is issued, the DASD fast write function begins when these resources become available. (For information on making these resources available to the subsystem, see [“SET CACHE”](#) on page 1422 and [“SET NVS”](#) on page 1588.)
3.  **Attention:** The FORCEOFF operand is intended only for recovery purposes and can result in a loss of data. It should only be used after SET DASDFW OFF has been issued to the device and the subsystem could not deactivate the DASD fast write function because of internal errors.

4. SET DASDFW OFF may produce delayed responses. See [“Delayed Responses to CP Commands”](#) on page 7 for more information.
5. This command may be issued to offline devices.
6. The caching function is always enabled on RAID (Redundant Array of Independent Disks) technology DASD such as an IBM DASD subsystem. For these types of DASD, the device accepts but ignores this command; the caching function cannot be modified by the user. The caching function for these types will always appear as ON in related query commands.

## Responses

Response 1:

The following response is displayed when you enter SET DASDFW ON and is repeated for each device specified on the command line.

```
rdev DASD fast write function has been activated.
```

If the originator of the command is not the system operator, the following response is issued to the system operator and is also repeated for each device specified on the command line.

```
rdev DASD fast write function was activated by userid.
```

### **rdev**

is the real device number of a device for which the DASD fast write function has been activated.

### **userid**

is the ID of the user who issued the command.

Response 2:

**Immediate Response:** If you enter SET DASDFW FORCEOFF, or if the DASD fast write function is deactivated immediately when you enter SET DASDFW OFF, the following response is repeated for each device specified on the command line.

```
rdev DASD fast write function has been deactivated.
```

### **rdev**

is the real device number of a device for which the DASD fast write function has been deactivated.

**Delayed Response:** If you issued the SET DASDFW OFF command and the DASD fast write function was not deactivated immediately, the following response is displayed:

```
Command started: SET DASDFW OFF rdev.
```

When the DASD fast write function is deactivated, the following response is issued:

```
rdev DASD fast write function has been deactivated.
```

Under certain circumstances, it is possible that a cache storage subsystem does not notify CP when it has deactivated the DASD fast write function. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing is ended after generating the following response:

```
Command results lost: SET DASDFW OFF rdev.
```

In this case the QUERY DASDFW command can be issued to determine the state of the DASD fast write function.

If errors occur during the asynchronous processing of the command that prevent the successful completion of the command, the following response is displayed:

```
Command failed: SET DASDFW OFF rdev.
```

**rdev**

is the real device number of a device for which the DASD fast write function failed to deactivate.

**Response to System Operator:** If the originator of the command is not the system operator, the following response is issued to the system operator when the DASD fast write function has been deactivated. This response is also repeated for each device specified on the command line.

```
rdev DASD fast write function was deactivated by userid.
```

**rdev**

is the real device number of a device for which the DASD fast write function has been deactivated.

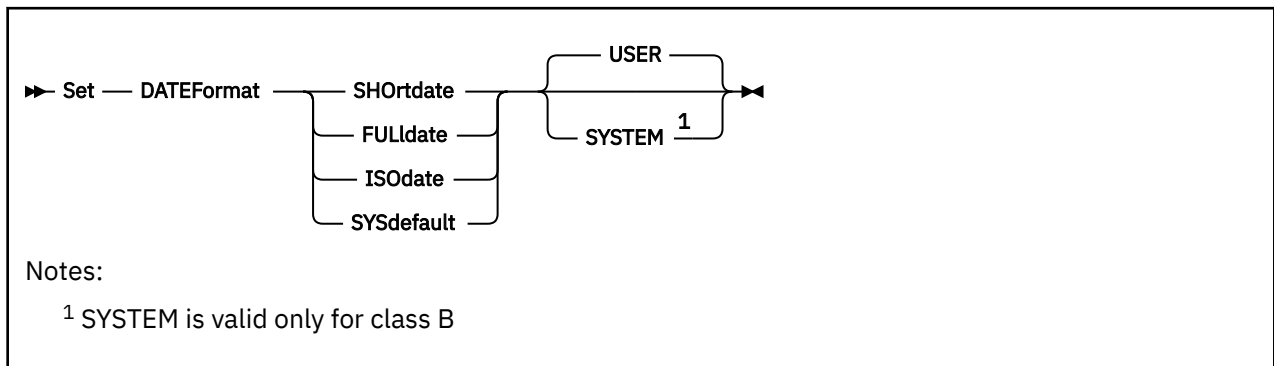
**userid**

is the ID of the user who issued the command.

**Messages**

- HCP006E Invalid device type - {rdev/vdev/ldev}
- HCP009E Invalid range - range
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP046E type rdev offline
- HCP319E Error processing subsystem command - cmd for DASD rdev
- HCP332E Invalid control unit type - rdev
- HCP1109E The command command cannot be processed for device rdev because the device is the secondary in a duplex pair.

## SET DATEFORMAT



### Authorization

Privilege Class: B, G

### Purpose

Use SET DATEFORMAT to set the default date format for commands which provide multiple date formats. Default date formats may be set for individual users or the entire system.

### Operands

#### SHOrtdate

specifies that dates will be displayed in *mm/dd/yy*, *mm/dd*, or *yy/mm/dd* format, where *mm* is the month, *dd* is the day of the month, and *yy* is the 2-digit year.

#### FULLdate

specifies that dates will be displayed in *mm/dd/yyyy* or *yyyy/mm/dd* format, where *mm* is the month, *dd* is the day of the month, and *yyyy* is the 4-digit year.

#### ISOdate

specifies that dates will be displayed in *yyyy-mm-dd* format, where *yyyy* is the 4-digit year, *mm* is the month, and *dd* is the day of the month.

#### SYSdefault

specifies that the default date format for this user will be set to the system-wide default.

#### USER

specifies that the default date format is being set for the user.

#### SYSTEM

specifies that the default date format is being set for the entire system.

### Usage Notes

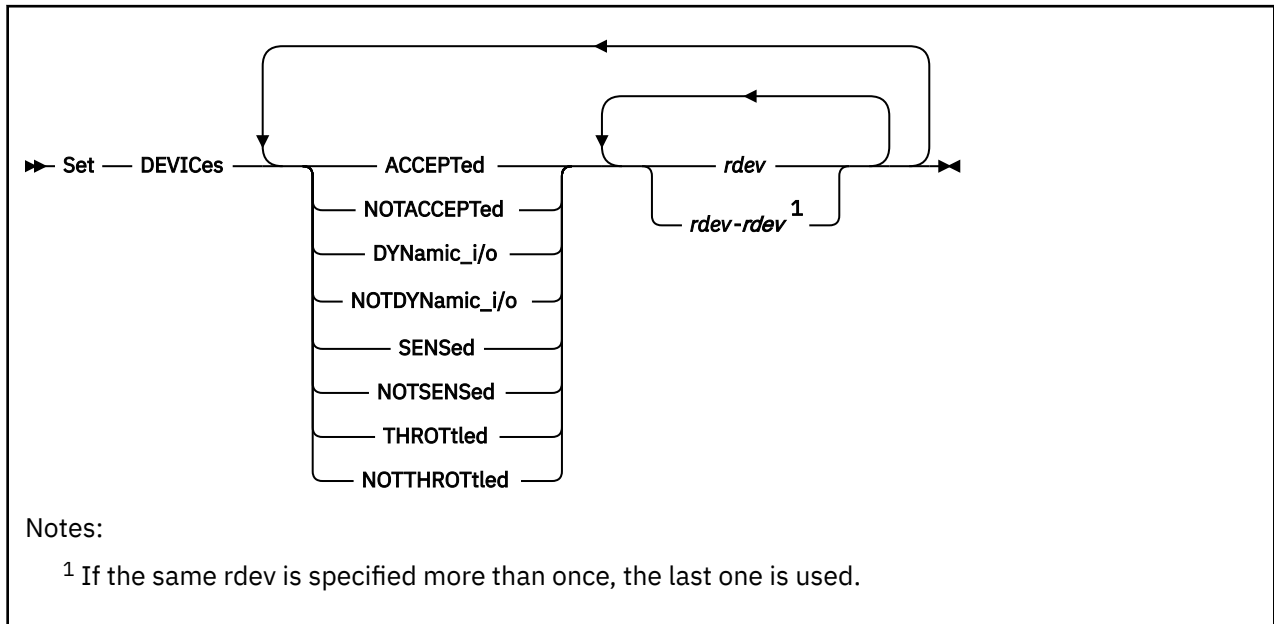
1. If SET DATEFORMAT is not specified for an individual user and there is no DATEFORMAT statement in the user's directory entry, the default date format for that user is the system-wide default.
2. SYSDEFAULT is not valid with SYSTEM.
3. If the default date format for a user is set to (or defaults to) the system-wide default date format, and the system-wide default is changed, the user continues to see the old system-wide default as the user default until the user either logs off and logs back on or issues SET DATEFORMAT SYSDEFAULT to switch to the new system-wide default.



**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP6704E Missing token at end of line

## SET DEVICES



### Authorization

Privilege Class: C

### Purpose

Use SET DEVICES to define or change the settings that CP will use if a device is added dynamically to the I/O configuration or is powered on after IPL.

**Note:** Before using this command, you must enable it by specifying a FEATURES ENABLE SET\_DEVICES statement in your system configuration file. See *z/VM: CP Planning and Administration* for more information on the [FEATURES Statement](#).

### Operands

#### ACCEPTed

tells CP to accept the specified device or devices.

#### *rdev*

#### *rdev-rdev*

is the real device number of the device or devices that you are affecting. The variable *rdev* must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single device, a range of devices, or any combination thereof.

#### NOTACCEPTed

tells CP not to accept the specified device or devices when the devices are dynamically added to VM from another partition. When VM is running second level, it prevents a device from being accepted when the device is attached to the virtual machine in which VM is running. If VM dynamically defines a device for its own partition, the NOTACCEPTed designation is overridden.

#### DYNamic\_i/o

tells CP to allow dynamic I/O changes on the specified device or devices.

**Note:** If you want to allow dynamic I/O changes on **any** devices, do not forget to specify FEATURES ENABLE DYNAMIC\_I/O in your system configuration file. This FEATURES operand enables or disables

dynamic I/O changes for your system's processor. So, if you do not enable dynamic I/O changes on the processor, CP prevents dynamic I/O changes on any individual device or devices.

#### **NOTDYNAMIC\_i/o**

tells CP not to allow dynamic I/O changes on the specified device or devices.

#### **SENSEd**

tells CP to use the information that is returned from a sense ID request to determine the device class or type.

#### **NOTSENSEd**

tells CP not to use the information that is returned from a sense ID request to determine the device class or type.

#### **THROTTled**

tells CP to limit (throttle) the rate of I/O coming from the specified device or devices.

#### **NOTTHROTTled**

tells CP not to limit (throttle) the I/O rate of the specified device or devices.

### **Usage Notes**

1. If multiple subchannel set support is enabled, DEVICES settings that are defined by the SET DEVICES command or the DEVICES system configuration statement apply to all subchannel sets. For example, if device 1234 is NOTACCEPTED, then so is device 11234. If device 01234 is NOTACCEPTED and device 11234 is added dynamically, it is ignored.

### **Responses**

Response 1:

To have CP accept a device, enter the following:

```
set devices accept 701
The Devices bit map was updated successfully
Ready;
```

Response 2:

To have CP not permit throttling for a range of devices, enter the following:

```
set devices notthrottled 700-710
The Devices bit map was updated successfully
Ready;
```

### **Messages**

- HCP002E Invalid operand - *operand*
- HCP643E Command not enabled in system configuration file

## SET DIALDROP

---



### Authorization

Privilege Class: G

### Purpose

Use SET DIALDROP to control when devices DIALED to your virtual machine are dropped.

### Operands

#### ON

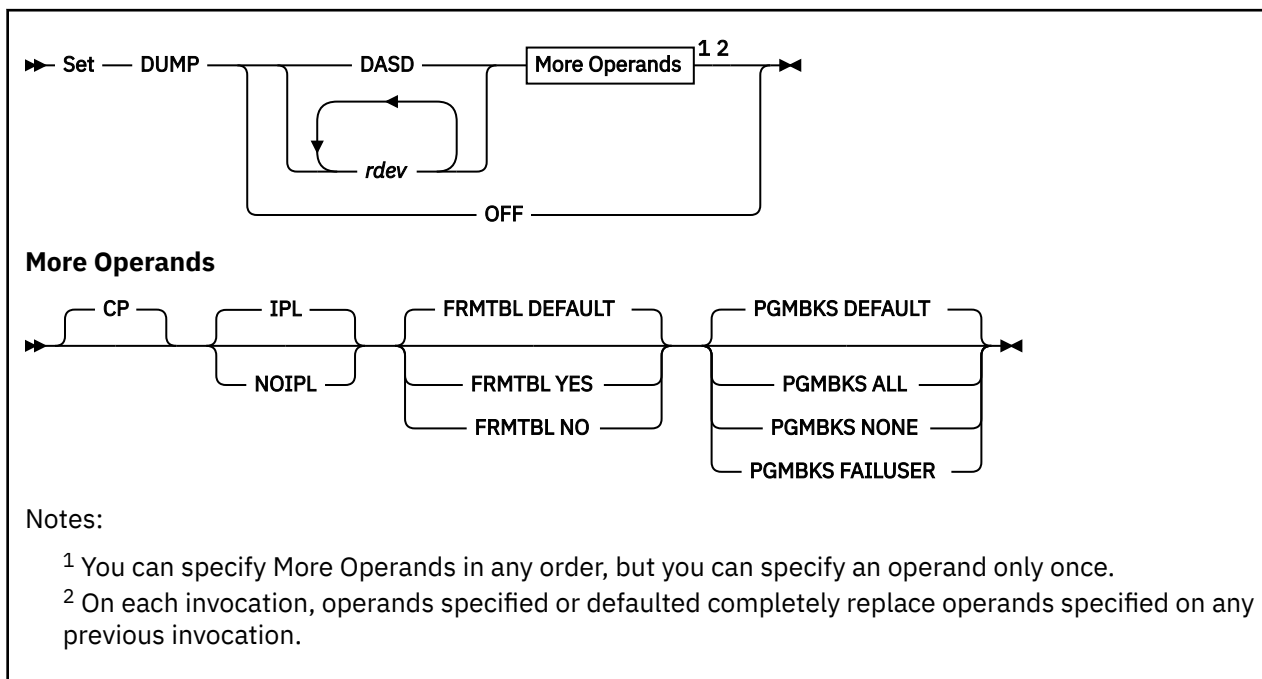
#### OFF

establishes whether devices DIALED to your virtual machine are dropped (ON) or stay connected (OFF) when a reset occurs (for example, when an IPL or SYSTEM CLEAR command is issued). Such a device is always dropped when an explicit DETACH or RESET command is issued for the associated virtual device. When you log on, DIALDROP is set ON.

### Messages

- HCP003E Invalid operand - *operand*
- HCP026E Operand missing or invalid

## SET DUMP



### Authorization

Privilege Class: B

### Purpose

Use SET DUMP to assign the unit or units to receive a system abend dump, to control the contents of a hard abend dump, and to specify whether the system should restart after dumping.

**Note:** Each invocation of SET DUMP completely replaces all settings from previous invocations of the command. Any option not specified on the new command is set to its default value.

### Operands

#### DASD

specifies that the system dump is written to one or more disks. The DASD dump space is released when the dump setting is changed. Clusters are allocated from spooling space on CP-owned DASD devices sufficient to hold a system abend dump.

#### rdev...

is the real device number, or a list of real device numbers of CP-owned DASD that contain spooling space to which the dump is written. A list of up to 32 real device numbers can be specified. The order in which you specify the DASD is the order in which they are searched for spooling space to be assigned as system abend dump space.

NVMe EDEVICES cannot be used to hold system hard abend dumps or snapdumps.

#### CP

is an obsolete keyword and no longer controls what memory is dumped. It is retained for compatibility.

**IPL  
NOIPL**

indicates whether the system is to be automatically restarted after a failure. At system initialization, the value is IPL; if you do not want your system automatically restarted, you must explicitly change the setting to NOIPL.

**OFF**

indicates that in the event a dump is attempted, either by a CP abend, a System Restart, or SNAPDUMP command, no dump will be created.

**FRMTBL DEFAULT****FRMTBL YES****FRMTBL NO**

specifies whether the full CP frame table is included in a hard abend dump. If you specify DEFAULT, the frame table is included in the cases for which IBM believes the frame table is needed to debug the problem. Otherwise, the dump includes a correlation table describing only the frames whose contents were dumped. YES forces inclusion of the frame table and NO forces the dump to have a correlation table. IBM recommends using the DEFAULT setting unless IBM support personnel instruct you to do otherwise in order to diagnose a specific problem.

**PGMBKS DEFAULT****PGMBKS ALL****PGMBKS NONE****PGMBKS FAILUSER**

specifies whether user PGMBKS are included in a hard abend dump. If you specify DEFAULT, PGMBKS are included in the cases for which IBM believes they are needed to debug the problem. ALL forces inclusion of all PGMBKS that aren't paged out, and NONE omits user PGMBKS. FAILUSER causes PGMBKS to be included for the guest that was running on the failing processor at the time of the abend, and for PGMBKS owned by the system. IBM recommends using the DEFAULT setting unless IBM support personnel instruct you to do otherwise in order to diagnose a specific problem.

PGMBKS owned by the system are always dumped, even if you specify PGMBKS NONE or PGMBKS FAILUSER. These PGMBKS are necessary in order to find data in address spaces owned by the system for certain VMDUMPTL commands.

**Usage Notes**

1. When the system is initialized, the dump device is set to DASD if there is enough spooling space to hold it. If there is not enough spooling space, a 'no dump unit' indicator is set. No dump will be generated.
2. When extending the amount of dump space during normal system execution, the system may be forced to set the 'no dump unit' indicator because there is insufficient contiguous SPOOL space.
3. The frame table and PGMBKS together constitute the vast majority of CP-owned frames, because they contain entries describing each 4K unit of real and virtual memory, respectively. The CP frame table is 1/128th the size of configured memory (8 GB per TB). PGMBKS take up 8K for each 1MB of virtual memory; though they are pageable, a PGMBK must remain resident as long as any page of that megabyte is resident. Omitting these structures from dumps when they are not likely to be needed for diagnosis dramatically reduces dump size and dumping time. With you specify the DEFAULT operand, CP determines whether or not to include the structures based on the abend code and failing module.
4. Specifying PGMBKS NONE/FAILUSER or FRMTBL NO does not affect the amount of space allocated for the dump. Dump space is allocated for a dump of maximum size because a SNAPDUMP command could occur at any time and could specify PGMBKS ALL and/or FRMTBL YES.
5. Use the OFF option only when you truly do not wish to receive a system dump. Use of this option stops the system dump capability until the command is reissued to specify an actual device.
6. When SET DUMP OFF has been specified, and the SET ABEND command was used to redefine a soft abend as HARD, an occurrence of that redefined soft abend will still cause a hard abend and a system re-IPL. But because of the SET DUMP settings, no dump will be generated. If the soft abend

was redefined as SNAPDUMP, the snapdump will not be generated because the only purpose of a snapdump is to create a dump.

7. When SET DUMP OFF is specified, the hard abend dump spool file is automatically closed. This is a class D spool file that is normally opened during system initialization and is owned by the user ID specified with the DUMP operand on the SYSTEM\_USERIDS configuration statement. It has a file type of "CPDUMP". Once closed, it can be reopened by issuing SET DUMP DASD or SET DUMP *rdev*.
8. See *z/VM: CP Planning and Administration* for information on [Allocating Space for CP Hard Abend Dumps](#) including recommendations from IBM.
9. Although CP limits the command length to 240 characters, not all terminal configurations allow that much input. It is possible that specifying all of the operands permitted by this command would exceed the input limits on some terminals. There are two ways to circumvent this limitation. First, it might be possible to change the line length of your terminal definition. For example, your terminal emulation software may permit you to define a terminal that has line 160 characters long, as opposed to the common line length of 80 characters. A second alternative is to create a simple REXX EXEC that contains the SET DUMP command that you wish to issue. This method has the added benefit of minimizing typographic errors because the REXX EXEC can be edited and rerun many times.

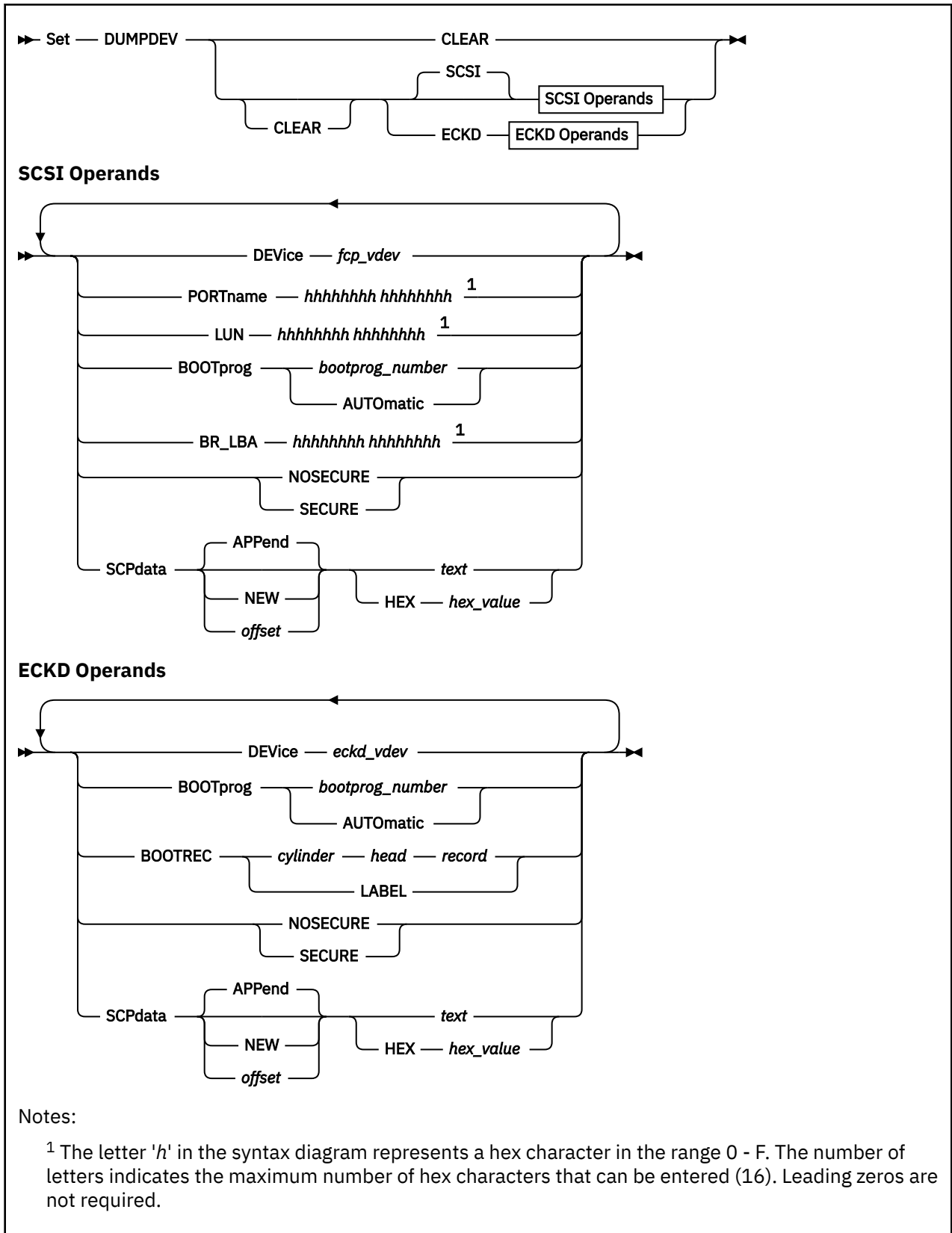
## Responses

The SET DUMP command responses are the same as the QUERY DUMP command responses. For more information, see the QUERY DUMP Responses section (see [“QUERY DUMP”](#) on page 887).

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP013E Conflicting option - *option*
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP263E Too many operands were supplied for this command.
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP847E Maximum system spool file limit exceeded
- HCP953E CP is unable to allocate system DASD dump space.
- HCP6005E Option *option* is not supported by z/VM.
- HCP9255I Excessive fragmentation of DASD dump space; unable to obtain additional clusters.
- HCP9259I No dump unit is set. No system abend dump will be generated. To assign a device, enter the SET DUMP command.

## SET DUMPDEV





## Authorization

Privilege Class: G

## Purpose

Use the **SET DUMPDEV** command to set parameters for a guest IPL with dump from a SCSI or ECKD disk. The parameters can identify the virtual device number and specify whether the IPL is a secure IPL. The parameters can identify the location of a system dumper program and information to pass to the program. The device that is IPLed is used to store the dump data.

The parameters that are set by using the **SET DUMPDEV** command are called DUMPDEV parameters and are used only for list-directed IPLs, which occur in the following cases:

- The **IPL** command specifies the *fcv\_vdev* operand and the DUMP operand.
- The **IPL** command specifies the DUMPDEV operand.

See usage note [“1” on page 1499](#).

The settings are temporary. The settings persist after an IPL but not after the user logs off. For more information, see [“IPL” on page 566](#).

## Operands

### CLEAR

The CLEAR operand clears all DUMPDEV parameters for all device types (SCSI and ECKD). Operands that follow the CLEAR operand are not cleared.

After DUMPDEV parameters are cleared, a guest IPL from SCSI or ECKD disk cannot be initiated until all required parameters for the SCSI or ECKD device type are set. The parameters that are required for each load device type are explained in usage note [“1” on page 1499](#).

### SCSI

The SCSI operand indicates that the DUMPDEV parameters are for IPL of a SCSI device. Other operands on the command must be operands for a SCSI device.

If you do not specify the SCSI or ECKD operand, the device type defaults to SCSI.

After the SCSI parameter is set, subsequent **SET DUMPDEV** commands cannot set ECKD parameters until a **SET DUMPDEV** command with the CLEAR operand is specified. Likewise, DUMPDEV parameters for an ECKD device must be cleared before you can set DUMPDEV parameters for a SCSI device.

### ECKD

The ECKD operand indicates that the DUMPDEV parameters are for IPL of an ECKD device. Other operands on the command must be operands for an ECKD device.

Because SCSI is the default device type, the ECKD operand must be specified on every command that sets DUMPDEV parameters for an ECKD device.

After the ECKD parameter is set, subsequent **SET DUMPDEV** commands cannot set SCSI parameters until a CLEAR operand is specified. Likewise, DUMPDEV parameters for a SCSI device must be cleared before you can set DUMPDEV parameters for an ECKD device.

### DEVICE *fcv\_vdev*

The DEVICE value (*fcv\_vdev*) specifies the virtual device number of the FCP device to IPL.

The DUMPDEV DEVICE parameter must be set before you can issue the **IPL DUMPDEV** command. A secure IPL with dump from a SCSI device can be initiated only by issuing the **IPL DUMPDEV** command.

If the **IPL** command specifies the *fcv\_vdev* operand, then the DUMPDEV DEVICE parameter must match that value or must not be set.

### DEVICE *eckd\_vdev*

The DEVICE value (*eckd\_vdev*) specifies the virtual device number of the ECKD device to IPL.

## SET DUMPDEV

The DUMPDEV DEVICE parameter must be set before you can issue the **IPL DUMPDEV** command. A secure IPL with dump from an ECKD device can be initiated only by issuing the **IPL DUMPDEV** command.

### **PORTname** *hhhhhhhh hhhhhhhh*

The PORTNAME value (*hhhhhhhh hhhhhhhh*) specifies the 8-byte fibre channel port name of the SCSI device that is IPLed. The value must be a hexadecimal value in the range 0 - FFFFFFFF FFFFFFFF. For values longer than eight hex characters, at least one separator blank is required after the eighth character.

If you do not set the PORTNAME parameter, it defaults to 0.

### **LUN** *hhhhhhhh hhhhhhhh*

The LUN value (*hhhhhhhh hhhhhhhh*) specifies the 8-byte logical unit number of the SCSI device that is IPLed. The value must be a hexadecimal value in the range 0 - FFFFFFFF FFFFFFFF. For values longer than eight hex characters, at least one separator blank is required after the eighth character.

If you do not set the LUN parameter, it defaults to 0.

### **BOOTprog** *bootprog\_number*

The *bootprog\_number* value specifies the boot program that is loaded from the SCSI or ECKD device that is IPLed. The value must be a decimal value in the range 0 - 30.

If you do not set the BOOTPROG parameter, it defaults to 0.

### **BOOTprog** **AUTOMATIC**

The BOOTPROG AUTOMATIC operand loads the first system dumper program (not an operating program) that is listed in the program table of the device that is IPLed.

If you do not set the BOOTPROG parameter, it defaults to 0.

### **BR\_LBA** *hhhhhhhh hhhhhhhh*

The BR\_LBA value (*hhhhhhhh hhhhhhhh*) specifies the logical-block address of the boot record of the SCSI device that is IPLed. The value must be a hexadecimal value in the range 0 - FFFFFFFF FFFFFFFF. For values longer than eight hex characters, at least one separator blank is required after the eighth character.

If you do not set the BR\_LBA parameter, it defaults to 0.

### **BOOTREC** *cylinder head record*

The BOOTREC values (*cylinder head record*) specify the boot record location on the ECKD device that is IPLed. Use hexadecimal values and separate the values by blank space. Leading zeros are not required. The following ranges are allowed when you specify the boot record location:

- Cylinder: 0 - FFFFFFFF
- Head (or track): 0 - F
- Record: 1 - FF

### **BOOTREC** **LABEL**

The BOOTREC LABEL operands specify that the boot record is the record that is identified in the volume label of the ECKD device that is IPLed.

If you do not set the BOOTREC parameter, it defaults to LABEL.

### **SECURE**

#### **NOSECURE**

The SECURE operand specifies that the IPL is a secure IPL.

The NOSECURE operand specifies that the IPL is not a secure IPL.

If you do not set the SECURE or NOSECURE parameter, the parameter defaults to NOSECURE.

The SECURE parameter is not valid unless the DEVICE parameter (virtual device number) is set.

**Note:** A secure IPL is required in the following situations:

- If a user's OPTION directory statement includes the SECUREIPLREQUIRED option, a secure IPL is required.
- If the program to be dumped was IPLed with the SECURE parameter, then the DUMPDEV parameters must include the SECURE parameter.

**SCPDATA ... text****SCPDATA ... HEX *hex\_value***

The SCPDATA keyword designates data to be passed to the program that is loaded for a list-directed IPL. If the program does not require any data, then the SCPDATA parameter is optional.

The HEX keyword indicates that the data (*hex\_value*) is UTF-8 encoded hex (characters 0-F). The hex string must be an even number of digits.

If the HEX keyword is not specified, the data is assumed to be EBCDIC text (code page 924).

Quotation marks are necessary only if the text includes blanks, mixed case, is a valid keyword (NEW or APPEND), or is all numeric (a valid offset).

Up to 4096 (4K) characters of data (text or hexadecimal) can be entered. The actual number of input characters might be less than 4096 depending on the translation to UTF-8. Because two hex characters are required to represent each UTF-8 data byte, the maximum number of UTF-8 data bytes that can be defined by using the HEX option is 2048 (1/2 of the 4K character input limit).

There is no default value for SCPDATA.

**NEW**

The parameter value that follows the NEW keyword replaces any previously specified SCPDATA data.

**APPEND**

The parameter value that follows the APPEND keyword is appended to the end of any previously specified SCPDATA data.

**offset**

The *offset* value indicates an offset of the previously specified SCPDATA data. The parameter value that follows the *offset* value overlays the previously defined SCPDATA data beginning at the specified offset. The *offset* value must be a value from 0 to the current length of the SCPDATA data.

**Usage Notes**

1. LOADDEV and DUMPDEV parameters are used for list-directed IPLs. [Table 34 on page 1499](#) shows which DUMPDEV parameters have default values that must be set (changed from the default) before each type of guest IPL with dump can be started. Other parameters might need to be set if the default values are not appropriate for your environment.

IPL type	DEVICE parameter	PORTNAME parameter	LUN parameter	SECURE parameter
IPL <i>fcv_vdev</i> DUMP	The DEVICE parameter is not set or matches <i>fcv_vdev</i> on the <b>IPL</b> command	Required	Required	The SECURE parameter is not allowed when the <i>fcv_vdev</i> operand is used
IPL <b>DUMPDEV</b> from SCSI device	Required	Required	Required	Required if the previous IPL used the SECURE parameter or if the SECUREIPLREQ option is specified in the OPTION directory statement
IPL <b>DUMPDEV</b> from ECKD device	Required	Not applicable	Not applicable	

## SET DUMPDEV

Note that a list-directed IPL with dump uses DUMPDEV parameters. A list-directed IPL without dump uses LOADDEV parameters. See [“IPL” on page 566](#) and [“SET LOADDEV” on page 1544](#).

2. To prepare for a secure IPL, the program that is loaded must be signed. The appropriate certificate must be loaded in the HMC and assigned to any LPAR on which the IPL might occur.
3. DUMPDEV parameters can be set only by using the **SET DUMPDEV** command.
4. The DUMPDEV settings of a user can be displayed by using the **QUERY DUMPDEV** command. See [“QUERY DUMPDEV” on page 890](#).
5. The initial **SET DUMPDEV** command or the first **SET DUMPDEV** command after a **SET DUMPDEV CLEAR** command establishes the device type (SCSI or ECKD) until another **SET DUMPDEV CLEAR** command is issued. Until another **SET DUMPDEV CLEAR** is issued, **SET DUMPDEV** commands with operands for a different device type are not valid. Because the default device type is SCSI, when you set an ECKD load device you must specify the ECKD operand on every **SET DUMPDEV** command issued. For example:

```
SET DUMPDEV CLEAR ECKD DEVICE 1234
SET DUMPDEV ECKD BOOTREC 0 0 5
```

6. After a DUMPDEV parameter is set, it can be changed by issuing the **SET DUMPDEV** command with a new value for the same parameter. It is not necessary to issue a **SET DUMPDEV CLEAR** command before you change a parameter value.

The one exception is the device type parameter (SCSI or ECKD). You must issue a **SET DUMPDEV CLEAR** command before you change the device type parameter.

7. The z/VM stand-alone dump utility does not support list-directed IPL from ECKD, and does not support secure IPL from SCSI or ECKD. Therefore, you cannot set the following DUMPDEV parameters to IPL z/VM stand-alone dump in a virtual machine:

- ECKD
- SECURE

More information about IPLing the z/VM stand-alone dump utility is available. See [The Stand-Alone Dump Utility in z/VM: CP Planning and Administration](#).

8. The default DUMPDEV value for the PORTNAME and LUN parameters is 0, which is not a valid device port name and not a valid device LUN. An IPL attempt that uses the default value for the PORTNAME or LUN parameter will fail.

### Examples

1. The following command identifies port a20, logical unit number 03 as the SCSI device, and passes the SCPDATA=x'74657374' to the system dumper program that is loaded during a guest IPL.

```
set dumpdev port a20 lun 03 scpdata hex 74657374
```

2. After the command in the previous example completes, the following command replaces '57' in the SCPDATA with '75'. The offset of 3 indicates where the data overlay begins. The resulting SCPDATA value is 74675374.

```
set dumpdev scpdata 3 hex 75
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP021E A device number was not supplied or it is invalid.
- HCP1613E An attempt was made to set both SCSI and ECKD parameters for a future IPL. These are conflicting parameters.
- HCP2768E Missing cylinder number

- HCP2768E Missing track number
- HCP2768E Missing record number
- HCP2813E CP is running on a processor that does not support guest secure IPL.
- HCP2813E One or more systems in your relocation domain do not support guest secure IPL.
- HCP2813E CP is running on a processor that does not support list-directed IPL from ECKD devices.
- HCP2813E One or more systems in your relocation domain do not support list-directed IPL from ECKD devices.
- HCP2813E CP is running on a processor that does not support BOOTPROG AUTOMATIC.
- HCP2813E One or more systems in your relocation domain do not support BOOTPROG AUTOMATIC.
- HCP2825E SCPDATA format conflicts with previous input
- HCP2826E SCPDATA contains invalid UTF-8 data: *data*
- HCP2827E SCPDATA offset exceeds current length of data (*length*)
- HCP2828E SCPDATA length exceeds maximum allowed (*maximum*)
- HCP2829E SCPDATA does not contain a valid hex string
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*
- HCP6706E Invalid number - *number*
- HCP6706E Invalid cylinder number - *cylinder*
- HCP6706E Invalid track number - *track*
- HCP6706E Invalid record number - *record*

## SET DYNAMIC\_I/O



### Authorization

Privilege Class: C

### Purpose

Use SET DYNAMIC\_I/O or SET DYNAMIC\_IO to enable or disable CP's ability to dynamically change the processor's I/O configuration.

### Operands

#### ON

tells CP to allow dynamic changes to the I/O configuration on this processor.

#### OFF

tells CP not to allow dynamic changes to the I/O configuration on this processor.

### Usage Notes

1. You can also allow or prevent dynamic I/O changes to this processor's I/O configuration using the ENABLE or DISABLE DYNAMIC\_I/O operands of the FEATURES statement in the system configuration file. For more information about the [FEATURES Statement](#), see *z/VM: CP Planning and Administration*.
2. After you enable dynamic I/O changes for the system (using either this command or the FEATURES statement), you can decide whether to allow dynamic I/O changes on a device by device basis using the DYNAMIC\_I/O operand of the SET DEVICES command or the DEVICES statement in the system configuration file. For more information about the SET DEVICES command, see [“SET DEVICES” on page 1490](#). For more information about the [DEVICES Statement](#), see *z/VM: CP Planning and Administration*.

### Responses

Response 1:

To allow dynamic I/O changes on the system, enter the following SET DYNAMIC\_I/O command:

```
set dynamic_i/o on
System is now enabled for dynamic I/O changes
Ready;
```

If CP is already allowing dynamic I/O changes when you issue the SET DYNAMIC\_I/O ON command, you see the following response:

```
set dynamic_i/o on
System is already enabled for dynamic I/O changes
Ready;
```

Response 2:

To prevent dynamic I/O changes on the system, enter the following SET DYNAMIC\_I/O command:

```
set dynamic_i/o off
System is now disabled for dynamic I/O changes
Ready;
```

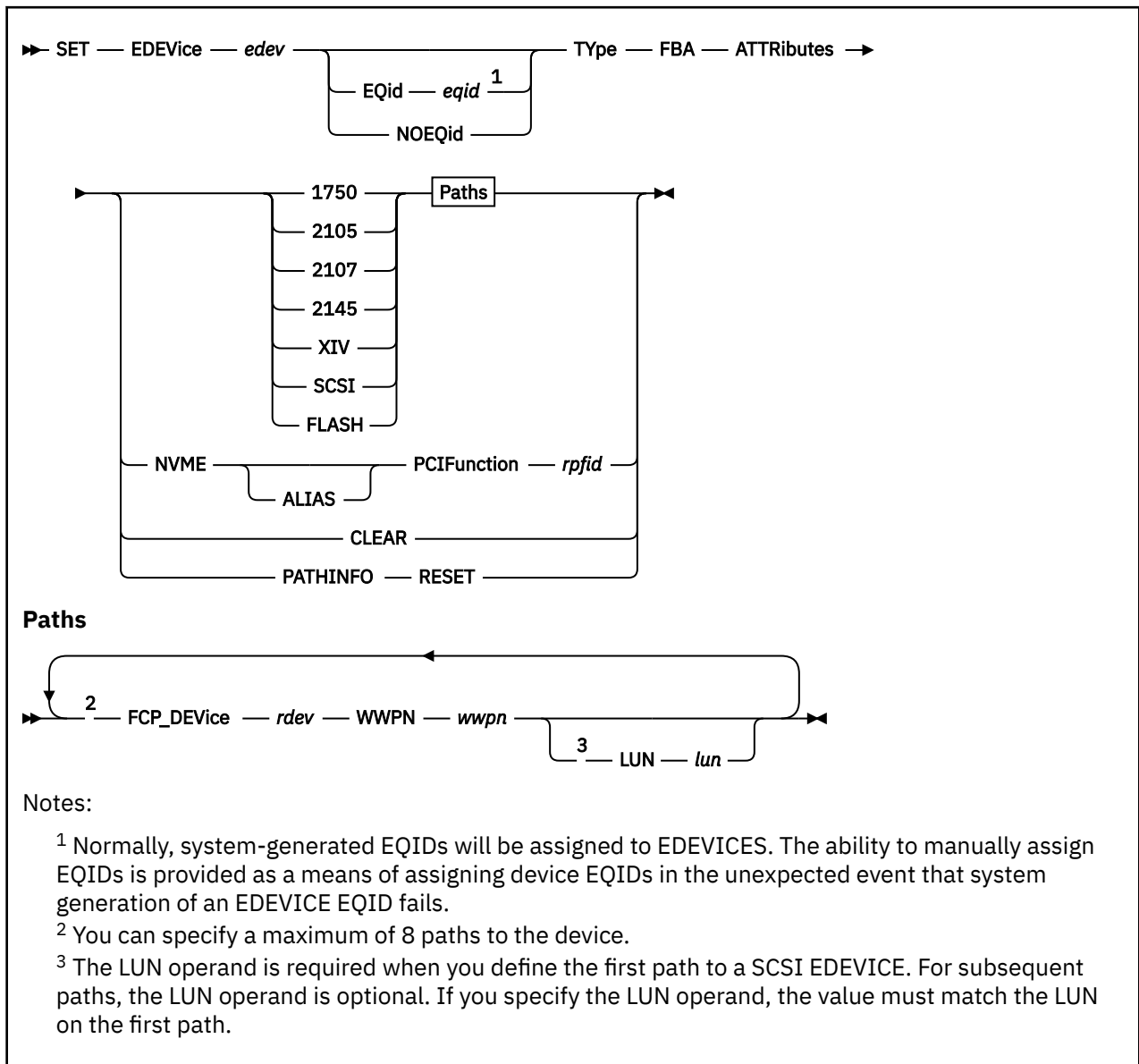
If CP is already preventing dynamic I/O changes when you issue the SET DYNAMIC\_I/O OFF command, you see the following response:

```
set dynamic_i/o off
System is already disabled for dynamic I/O changes
Ready;
```

## Messages

- HCP026E Operand missing or invalid
- HCP643E Command not enabled in system configuration file
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Cannot find configuration token in the channel subsystem
- HCP6815E Command failed in the channel subsystem
- HCP6816E Dynamic I/O changes are not allowed because LPAR information is not available
- HCP6850E Cannot disable dynamic I/O changes A dynamic I/O change is in progress
- HCP6850E Cannot disable dynamic I/O changes System is in configuration mode

## SET EDEVICE



### Authorization

Privilege Class: B

### Purpose

Use the SET EDEVICE command to define, modify, or clear the definition of an emulated device that represents a real SCSI device or a real NVMe device that is connected to a PCIe adapter.

**Note:** The device number of an emulated device must not conflict with the device number of a real device.

### Operands

#### *edev*

specifies the device number that is associated with the emulated device. The *edev* number corresponds directly with an *rdev* number. Thus, a new *edev* number cannot be defined where an *rdev*



number already exists. An *edev* in the active configuration can be entered as a 4-digit hexadecimal number between X'0000' and X'FFFF'. For an *edev* in the active or standby configuration, the device number can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set of the device defined, modified or cleared.

**EQid eqid**

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1 to 8 alphanumeric characters. If the CLEAR option is specified, the EQID will be ignored. The EQID must be unique across all devices on the system.

**NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID. If the CLEAR option is specified, the NOEQID option is not necessary.

**TYPe**

specifies the emulated device type, selected by the immediately following parameter.

**FBA**

identifies the emulated device as an FBA DASD.

**ATTRIBUTES**

specifies the name of an attribute set to be associated with the real device.

**1750**

specifies that IBM 1750 device attributes are to be used for the real device that is being emulated. This attribute should be used for all DS6000 hardware. See Usage Note [“6” on page 1507](#) for more information.

**2105**

specifies that IBM 2105 device attributes are to be used for the real device that is being emulated. This attribute should be used for all Enterprise Storage Server (ESS) hardware.

**2107**

specifies that IBM 2107 device attributes are to be used for the real device that is being emulated. This attribute should be used for all DS8000 hardware.

**2145**

specifies that IBM 2145 device attributes are to be used for the real device that is being emulated. This attribute should be used for all SAN Volume Controller hardware, including Storwize® models. See Usage Note [“7” on page 1507](#) for more information.

**XIV**

specifies that IBM XIV® device attributes are to be used for the real device that is being emulated.

**SCSI**

specifies that general SCSI device attributes are to be used for the real device that is being emulated.

**Note:** When specifying this value, you should exercise caution in defining more than one path to the device. Be sure that the device actually supports multiple paths. Defining multiple paths to a device that does not support multiple paths could result in data-integrity problems on the device.

**FLASH**

specifies that IBM FLASH device attributes are to be used for the real device that is being emulated. This attribute is used for IBM Flash Systems hardware and does not require a SAN Volume Controller (2145) in use.

**NVME**

specifies that NVMe device attributes are to be used for the device that is emulated. This attribute is used for NVMe devices that are connected via PCIe functions.

**FCP\_DEVICE rdev**

specifies the real device number of the FCP device to be used for a specific path to a SCSI device. The *rdev* must be a 1- to 4-digit hexadecimal number between X'0000' and X'FFFF'.

**WWPN wwpn**

specifies the world wide port name for a specific path to a SCSI device. The *wwpn* must be a 16-digit hexadecimal number between X'0000000000000000' and X'FFFFFFFFFFFFFFFF'.

## SET EDEVICE

**Note:** If fewer than 16 digits are specified for the *wwpn*, the number will be padded with leading zeroes to make it a 16-digit number.

### LUN *lun*

specifies the logical unit number for a specific path to a SCSI device. The *lun* must be a 16-digit hexadecimal number between X'0000000000000000' and X'FFFFFFFFFFFFFF'.

**Note:** If fewer than 16 digits are specified for the *lun*, the number will be padded with leading zeroes to make it a 16-digit number.

### ADD PATH

indicates that one or more additional paths are to be added to the real SCSI device.

### DELeTe PATH

indicates that one or more paths are to be deleted from the real SCSI device.

### ALIAS

specifies that an alias device is to be defined.

### PCIFunction *rpfid*

is the ID of a real PCI function that is associated with an NVMe adapter. Specify the ID as eight hexadecimal digits. You can omit leading zeros. If an EDEVICE is already associated with the PCI function, another EDEVICE is defined. One base device can be defined for roughly each one terabyte of device capacity. A combination of up to 127 base and alias devices can be defined for a single NVMe PCIe function.

### CLEAR

specifies that the information in the real device control block (RDEV) and the emulated device control block (EDEV) associated with the specified emulated device is to be cleared such that the emulated device can no longer be used without another SET EDEVICE command to redefine it.

The CLEAR operand is meaningless for an NVMe EDEVICE and is rejected if used with an NVMe EDEVICE.

### PATHINFO RESET

clears the current error and failure counters associated with each path. Also resets the timestamps and reason fields.

## Usage Notes

1. When defining emulated devices to represent real SCSI devices, there should be a one-to-one relationship between an emulated device and a real SCSI device. All paths defined for an emulated device should represent paths to the same real SCSI device. If the paths for one emulated device are associated with more than one real SCSI device or if more than one emulated device is associated with the same real SCSI device, data integrity problems could occur.
2. Path validation for an emulated device occurs when the emulated device is varied online or when a path is added to an online emulated device. Any invalid path will be deleted from the EDEV. A path is considered to be invalid for any of the following reasons:
  - The device specified by the FCP\_DEVICE parameter does not exist.
  - The device specified by the FCP\_DEVICE parameter is not an FCP device.
  - The device specified by the FCP\_DEVICE parameter is an offline FCP device.
  - The device specified by the FCP\_DEVICE parameter is dedicated to a virtual machine.
  - The value specified by the WWPEN parameter is not a valid world wide port name in your configuration.
  - The value specified by the LUN parameter is not a valid logical unit number for the specified world wide port name.
3. You must use the VARY command to take an emulated device offline before you can clear it.
4. The ADD PATH and DELETE PATH parameters are only valid when the command is issued to a previously defined emulated device, but cannot be used on a command issued to a cleared emulated device.

5. The last path to an emulated device cannot be deleted except via the CLEAR parameter.
6. Devices (LUNs) on an IBM 1750 (DS6000) storage controller can be configured through any of the I/O ports on the controller, but the access times for a given device through different ports can be different. For a given device, half of the ports will provide faster access times than the other half of the ports. The channel paths connected to the ports with faster access are called "preferred paths" for the device, and the remaining channel paths are called "non-preferred paths".

Previous releases of z/VM required knowledge of the preferred/non-preferred value for each path on a SET EDEV command when using the 1750 attribute. Now, z/VM is able to determine during device initialization which paths are preferred and which are not, so operands on the SET EDEV command that explicitly state the preferred/non-preferred value for such a device will be ignored. Preferred paths will still be given preference over non-preferred paths during the path selection for an I/O request.

7. When using an IBM 2145, a host that has a LUN mapping for a virtual disk will normally be able to access that virtual disk via any of the ports connected to either of the two nodes in the I/O group supporting that virtual disk. Each virtual disk has a set of preferred ports. These ports will give slightly better performance than the ports that are non-preferred. By default, the cache component assigns ownership of even numbered virtual disks to one node of a caching pair and the ownership of odd numbered virtual disks to the other node. To provide flexibility, the ownership for a given virtual disk can be explicitly assigned to a given node when the virtual disk is created. A node that is explicitly assigned as an owner of a virtual disk is known as the preferred node.

Previous releases of z/VM required knowledge of the preferred/non-preferred value for each path on a SET EDEV command when using the 2145 attribute. Now, z/VM is able to determine during device initialization which paths are preferred and which are not, so operands on the SET EDEV command that explicitly state the preferred/non-preferred value for such a device will be ignored. Preferred paths will still be given preference over non-preferred paths during the path selection for an I/O request.

8. If an EQID is specified along with the CLEAR parameter, the EQID is ignored and the EDEV is cleared without setting an EQID. Similarly, if an EQID is specified with the PATHINFO RESET parameters, the EQID is ignored and the path information is reset without setting an EQID.
9. When setting an EQID for an EDEV, no other device can have the same EQID assigned to it. Furthermore, when the device is brought online, if a system-generated EQID can be assigned to the device, the user-defined EQID will be removed and the system-generated ID will be used.
10. Normally, system-generated EQIDs will be assigned to EDEVICES. The ability to manually assign EQIDs is provided as a means of assigning device EQIDs in the unexpected event that system generation of an EDEVICE EQID fails.
11. When using IBM Flash Systems and specifying the FLASH attribute, a SAN Volume Controller should not exist in the configuration. In this case a round-robin path algorithm is used. However, if a SAN Volume Controller is configured with Flash Systems hardware, the 2145 attribute must be specified instead of the FLASH attribute, because a preferred path algorithm is used.
12. For system attached paging EDEVICES, z/VM has an optimized path for issuing SCSI I/O requests directly without any simulation overhead. For non-paging EDEVICES, there is simulation overhead incurred with every I/O request. Therefore, it is recommended that EDEVICES only be used for disks which are accessed infrequently, such as a system disk used to store z/VM kernel images. z/VM has industry leading virtualization support for FCP subchannel attachment, which should be considered the primary approach for configuring any I/O intensive environments.
13. When configuring a single LUN on an XIV, SVC (2145), or a device incorporating SVC technology (such as V7000, V840, and V9000) as an EDEVICE, each associated FCP (source) subchannel should be configured to only one target port (WWPN) of the storage system. In addition, it is recommended that each (source) FCP subchannel be defined on a different (source) FCP channel path (or CHPID). This provides for best LUN availability in the event of a link failure.
14. Due to hardware differences between a Flash device and other SCSI devices, the FLASH attribute must be specified in order for an EQID to be generated. Otherwise, the device cannot be used in an SSI environment. The exception is: if the FlashSystem is used as an IPL device, the attributes FLASH, XIV, and SCSI will be recognized as valid for generating an EQID.

## SET EDEVICE

15. Specifying a 5-digit hexadecimal device number on defining an EDEV will place it into the active configuration as well, if there is not already a matching device number active.
16. When you vary on the device, the device attribute is checked dynamically. If the wrong value was set, the ATTR value will be updated to the correct device attribute. Any further paths added or deleted using the SET EDEVICE command will require specifying the correct ATTR operand. When setting the ATTR operand for non-IBM devices, the above behavior is ignored, and the specified ATTR value will be used as the specified device attribute for the emulated device.
17. When the ADD PATH or DELETE PATH option is specified, the value set for the ATTR operand is ignored, and the attribute of the existing EDEV is used to add or delete the path instead.

### Responses

Response 1:

To define an emulated device 8181 for a 2105 SCSI device with one path, enter the following:

```
set edev 8181 type fba attr 2105
  fcp_dev 900 wwpn 200400A0B80BA987 lun 0002000000000000
EDEV 8181 was created.
Ready;
```

Response 2:

To define an emulated device 8181 for a 2105 SCSI device with one path, and an EQID of ABCDEFGH, enter the following:

```
set edev 8181 eqid ABCDEFGH type fba attr 2105
  fcp_dev 900 wwpn 200400A0B80BA987 lun 0002000000000000
EDEV 8181 was created.
Ready;
```

Response 3:

To define an emulated device 8181 for a 2105 SCSI device with one path, and remove any previously existing EQID for the EDEV, enter the following:

```
set edev 8181 noeqid type fba attr 2105
  fcp_dev 900 wwpn 200400A0B80BA987 lun 0002000000000000
EDEV 8181 was created.
Ready;
```

Response 4:

To define an emulated device 4343 for a general SCSI device with two paths, enter the following:

```
set edev 4343 type fba attr scsi
  fcp_dev 800 wwpn 200400A0B80BA687 lun 0001000000000000
  fcp_dev 801 wwpn 200400A0B80BA688 lun 0001000000000000
EDEV 4343 was created.
Ready;
```

Response 5:

To add a path to an existing emulated device 8181, enter the following:

```
set edev 8181 type fba attr 2105 add path
  fcp_dev 901 wwpn 200400A0B80BA988 lun 0002000000000000
EDEV 8181 was modified.
Ready;
```

Response 6:

To delete a path from an existing emulated device 4343, enter the following:

```
set edev 4343 type fba attr scsi delete path
  fcp_dev 800 wwpn 200400A0B80BA687 lun 0001000000000000
```

```
EDEV 4343 was modified.
Ready;
```

## Response 7:

To clear an existing emulated device 4343, enter the following:

```
set edev 4343 clear
EDEV 4343 was cleared.
Ready;
```

## Response 8:

To define an emulated device 5004 for a 2145 device with two paths where the LUN is optional on the second path, enter the following:

```
set edev 5004 type fba attr 2145
  fcp_dev 800 wwpn 200400A0B80BA687 lun 0002000000000000
  fcp_dev 801 wwpn 200400A0B80BA688
EDEV 5004 was created.
Ready;
```

## Response 9:

To reset the error and failure information displayed with the PATHINFO option, enter the following:

```
set edev 400 pathinfo reset
```

The path information should now be cleared:

```
query edev 400 pathinfo
EDEV 0400 TYPE FBA ATTRIBUTES 2107 ALUA SUPPORTED: YES
PATHS:
FCP_DEV: 1D1D WWPN: 5005076306081588
LUN: 4064401600000000 ALUA OPTIMIZED: NO
SELECTED: 0
ADAPTER ERRORS: 0
SOFTWARE: 0
HARDWARE: 0
TRANSPORT: 0
OTHER: 0
TIMEOUT ERRORS: 0
SCSI ERRORS: 0
RESERVATION: 0
QUEUE FULL: 0
DEVICE BUSY: 0
ACA ACTIVE: 0
ABORT TASK: 0
CHECK CONDITIONS: 0
MEDIUM: 0
HARDWARE: 0
NOT READY: 0
ABORTED: 0
OTHER: 0
LAST ERROR TIME: NONE
LAST ERROR TYPE: NONE
LAST FAILURE TIME: NONE
LAST FAILURE REASON: NONE
```

The counters are reset to 0 and the error and failure types are set to NONE.

## Response 10:

The following commands and responses show the definition of emulated base devices (0F80, 0F81) and emulated alias devices (0F90, 0F91, 0F92) for a real PCIe function 00000071 that is associated with an NVMe adapter.

```
set edev f80 type fba attr nvme pcif 00000071
EDEV 0F80 was created.

vary on f80
```

## SET EDEVICE

```
0F80 varied online

attach f80 system
DASD 0F80 ATTACHED TO SYSTEM MYNVME HYPERPAV BASE

set edev f81 type fba attr nvme pcif 00000071
EDEV 0F81 was created.

vary on f81
0F81 varied online

attach f81 system
DASD 0F81 ATTACHED TO SYSTEM MYNV2 HYPERPAV BASE

set edev f90 type fba attr nvme alias pcif 00000071
EDEV 0F90 was created.

vary on f90
0F90 varied online

attach f90 system
DASD 0F90 ATTACHED TO SYSTEM          HYPERPAV ALIAS

set edev f91 type fba attr nvme alias pcif 00000071
EDEV 0F91 was created.

vary on f91
0F91 varied online

attach f91 system
DASD 0F91 ATTACHED TO SYSTEM          HYPERPAV ALIAS

set edev f92 type fba attr nvme alias pcif 00000071
EDEV 0F92 was created.

vary on f92
0F92 varied online

attach f92 system
DASD 0F92 ATTACHED TO SYSTEM          HYPERPAV ALIAS
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP040E Device *rdev* does not exist
- HCP040E PCI function *rpfid* does not exist.
- HCP048E A unique EQID cannot be generated for EDEV *edev* based upon the hardware information provided.
- HCP048E Specified EQID already assigned to a different device.
- HCP377E Real PCI function *rpfid* not NVMe
- HCP377E Real PCI function *rpfid* not configured
- HCP377E Real PCI function *rpfid* already in use
- HCP377E Error *code* enabling real PCI function *rpfid*.
- HCP377E Real PCI function *rpfid* is not enabled
- HCP377E Error disabling real PCI function *rpfid*
- HCP377E Error *code* storing NVMe *rpfid* PCI Function Controls.
- HCP377E Error *code* registering NVMe *rpfid* I/O Translation parameters.
- HCP377E Unable to allocate NVMe *rpfid* PCI Function address space.
- HCP377E Error *code* loading or storing NVMe *rpfid* PCI Function data
- HCP377E NVMe *rpfid* PCI Function class code *code* is not recognized

- HCP377E Error *code* registering NVMe *rpfid* adapter interruptions
- HCP377E Error *code* refreshing NVMe *rpfid* PCI translations
- HCP377E Error *code* submitting NVMe *rpfid* command
- HCP1014E A required option is missing - *option*
- HCP2777I Operand *operand* has been ignored.
- HCP6704E Missing token at end of line
- HCP6760E Device *rdev* is not offline
- HCP6870E Device *rdev* is attached to a guest
- HCP8701I Path FCP\_DEV *rdev* WWPN *wwpn* LUN *lun* was not defined to EDEV *rdev* because the maximum amount of paths are already defined.
- HCP8701I Path FCP\_DEV *rdev* WWPN *wwpn* LUN *lun* was not deleted because it is the last path associated with EDEV *rdev*.
- HCP8701I Path FCP\_DEV *rdev* WWPN *wwpn* LUN *lun* was not deleted because it is not a path associated with EDEV *rdev*.
- HCP8701I Path FCP\_DEV *rdev* WWPN *wwpn* LUN *lun* is a duplicate path for EDEV *rdev* and was not added due to the previous definition.
- HCP8702E Device *rdev* is defined as a real device.
- HCP8702E Device *rdev* is defined as an emulated device.
- HCP8702E Device *rdev* is not a defined emulated device.
- HCP8703E The type specified does not match the type that is already associated with EDEV *rdev*.
- HCP8703E The attributes name specified does not match the attributes name that is already associated with EDEV *rdev*.
- HCP8703E The LUN *lun* specified does not match the LUN that is already associated with EDEV *rdev*.
- HCP8704E EDEV *rdev* was not modified because the device could not be unloaded.
- HCP8704E EDEV *rdev* was not cleared because the device could not be unloaded.
- HCP8704E Device *rdev* is an NVMe emulated device and cannot be cleared.

## SET EMSG



### Authorization

Privilege Class: G

### Purpose

Use SET EMSG to control error message handling. The initial setting for EMSG is installation dependent and is defined in the system configuration file.

### Operands

#### ON

specifies that both error code and error message text are displayed on your display screen.

#### CODE

specifies that only the error code is displayed.

#### IUCV

specifies that both the error code and text are to be passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the message is handled as if SET EMSG ON had been issued.

#### OFF

specifies that no error messages are displayed.

#### TEXT

specifies that only the message text is displayed.

### Usage Notes

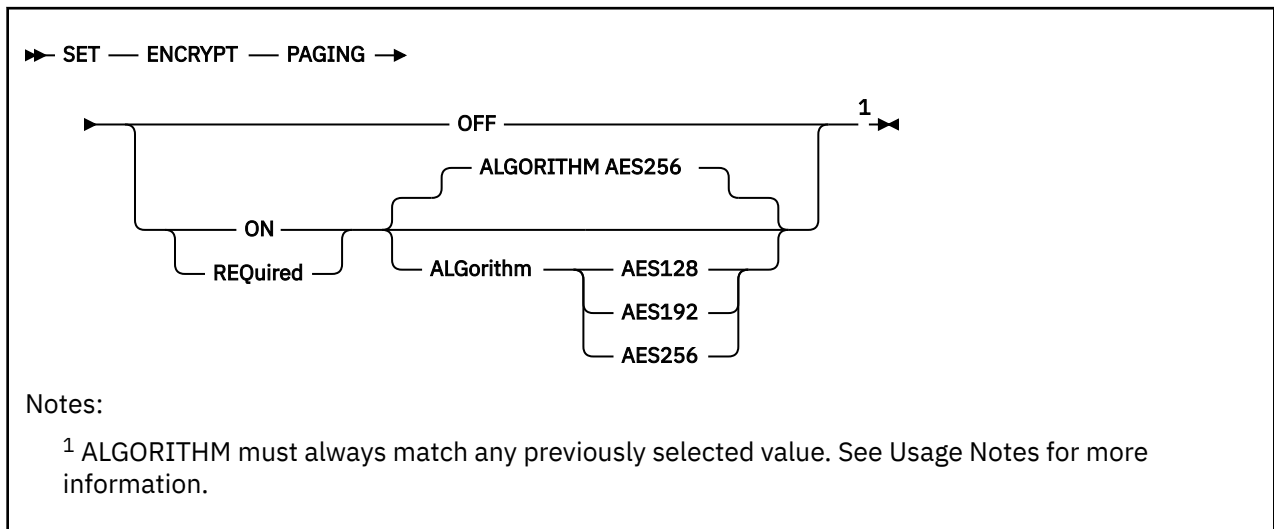
1. If you specify the TEXT option, the number of the message is not displayed. The lack of a message number makes it difficult to locate the message in the [z/VM: CP Messages and Codes](#). Therefore, use the TEXT option with caution.
2. The full message is spooled (code and text) if SET EMSG is set OFF; otherwise an error message is spooled according to the setting.
3. CMS recognizes EMSG settings for all error (E) and information (I) messages, but ignores the EMSG setting and displays the complete message (error code and text) for all other system messages.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid



## SET ENCRYPT



### Authorization

Privilege Class: A

### Purpose

Use SET ENCRYPT to enable or disable host level encryption for the z/VM hypervisor if the appropriate hardware support is available to the LPAR.

### Operands

#### PAGing

indicates the hypervisor function modified by this command is CP paging of guest memory and virtual-disk-in-storage (VDISK).

#### OFF

disables the host encryption for the hypervisor function indicated. You cannot use this operand if the ENCRYPT value was previously set to REQUIRED (via either the SET ENCRYPT command or the ENCRYPT configuration statement).

#### ON

enables host encryption for the hypervisor function indicated, if appropriate hardware is available. You cannot use this operand if the ENCRYPT value was previously set to REQUIRED (via either the SET ENCRYPT command or the ENCRYPT PAGING system configuration statement).

#### REQUIRED

enables host encryption for the hypervisor function indicated, if appropriate hardware is available, and locks this setting until the next system IPL. That is, once you set the ENCRYPT value to REQUIRED, the SET ENCRYPT command cannot be used for this CP function.

#### ALGorithm

Specifies the symmetric encryption cipher to be used by this host function. All algorithms currently supported require a specific level of CPACF (hardware feature 3863) to be enabled for the system. For more information, see *z/VM: Migration Guide*.

## SET ENCRYPT

### AES128

### AES192

### AES256

indicates the Advanced Encryption Standard (AES) algorithm is to be used for this hypervisor function, in Cipher Block Chaining (CBC) mode. The key length in bits determines the strength of the encryption to be performed.

## Usage Notes

1. A value for ALGORITHM can be selected only when ENCRYPT PAGING is enabled for the first time. The algorithm can be set either with the SET ENCRYPT command or with the ENCRYPT statement in the system configuration file. Once an algorithm is selected, it cannot be changed without a system IPL.
2. The algorithm must always match any previously selected value. SET ENCRYPT returns an error if a different algorithm is requested.
3. The system operator receives an informational message after a successful SET ENCRYPT command if there is a change in encryption status.
4. Caution should be exercised when using the REQUIRED option, because after using this option further changes are not permitted until the next system IPL. For more information, see [Pervasive Encryption for z/VM](#) in *z/VM: CP Planning and Administration*.
5. Enabling encryption will increase CPU utilization relative to the strength of the encryption algorithm selected. For more information, see [Major Factors Affecting Performance](#) in *z/VM: Performance*.

## Responses

Response 1:

```
Encrypt Paging set on to algorithm AES192
Encrypt Paging Settings:
  Currently: On AES192
  At IPL: Off
```

This response is issued when ON is specified with a valid algorithmic value.

Response 2:

```
Encrypt Paging set off
Encrypt Paging Settings
  Currently: Off
  At IPL: Off
```

This response is issued when encryption has never been enabled.

Response 3:

```
HCPENC1394I Encryption of paging changed from {on|off} to {on|off|required},
[with algorithm AESnnn,] by user userid
```

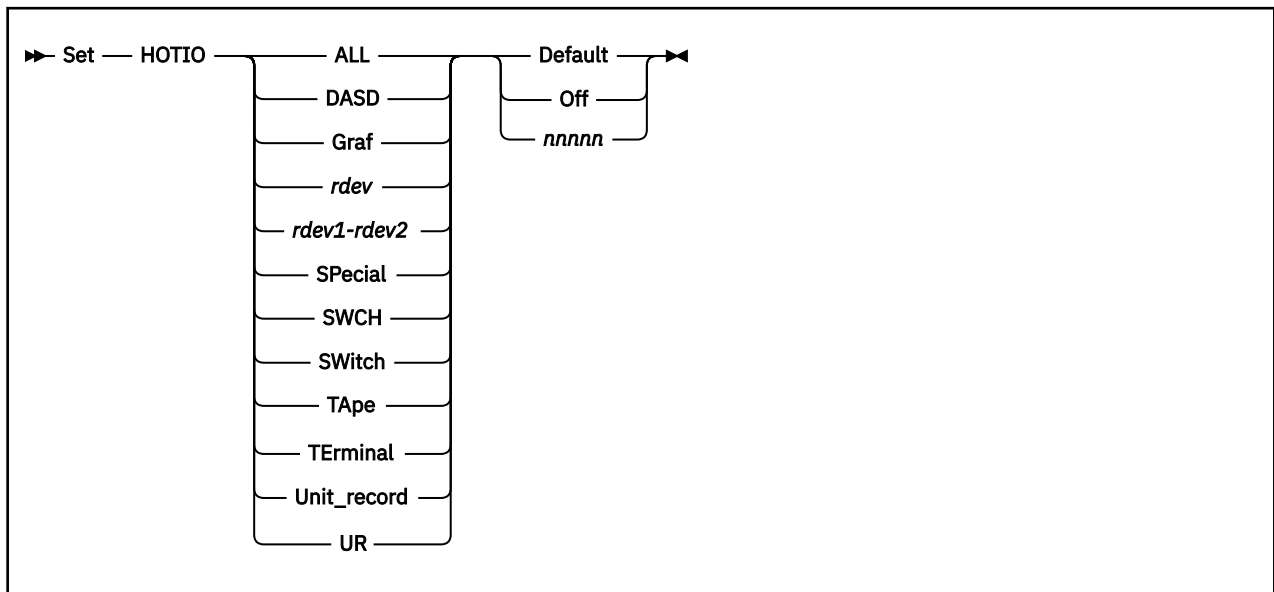
This message is issued to the primary system operator when a user ID other than the system operator uses SET ENCRYPT to change the state of host-level encryption of a particular function. This message does not require operator intervention, but should be noted for auditing and automation purposes.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP1390E Encrypt Paging cannot be enabled due to missing hardware support
- HCP1391E Encryption algorithm previously set to *algorithm*; no change made

- HCP1392E Encrypt Paging is required; no change made
- HCP1394I Encryption of paging changed from *state* to *state*, [with algorithm *algorithm*,] by user *userid*
- HCP2768E Missing algorithm type

## SET HOTIO



### Authorization

Privilege Class: B

### Purpose

Use SET HOTIO to change the maximum number of consecutive, unsolicited interrupts per second that CP should allow from an I/O device (the hot I/O rate) before refusing to accept input from it. For information on how to display the hot I/O rate settings for your system, see [“QUERY HOTIO” on page 938](#).

### Operands

#### ALL

tells CP to change the hot I/O rate for all the devices in the active I/O configuration.

#### DASD

tells CP to change the hot I/O rate for all the DASDs in the active I/O configuration.

#### Graf

tells CP to change the hot I/O rate for all the graphic display devices in the active I/O configuration.

#### rdev

tells CP to change the hot I/O rate for a device at a specific real device number in the active I/O configuration. The variable *rdev* must be a hexadecimal number between X'0000' and X'FFFF'. If you specify a real device number that does not exist, CP issues an error message and ignores your command.

#### rdev1-rdev2

tells CP to change the hot I/O rate for devices in a range of specific real device numbers in the active I/O configuration. The variable *rdev* must be a hexadecimal number between X'0000' and X'FFFF'. The first real device number in the range must be less than the second real device number; otherwise, CP issues an error message and ignores your command. If you specify a range of real device numbers that do not exist, CP issues an error message and ignores your command. However, if there are some valid real device numbers in the range, CP accepts the command.

**SPecial**

tells CP to change the hot I/O rate for all the special devices (3088s, CTCAs, 37x5s) in the active I/O configuration.

**SWCH****SWitch**

tells CP to change the hot I/O rate for all the switching devices in the active I/O configuration.

**TApe**

tells CP to change the hot I/O rate for all the tape drive devices in the active I/O configuration.

**TErминаl**

tells CP to change the hot I/O rate for all the terminals in the active I/O configuration.

**Unit\_record****UR**

tells CP to change the hot I/O rate for all the unit record devices in the active I/O configuration.

**Default**

tells CP to use the default rate of 16 unsolicited interrupts per second for the specified device or devices.

**Off**

turns hot I/O detection off and tells CP to accept all unsolicited interrupts for the specified device or devices.

**Warning!**

We do not recommend that you specify the OFF operand to turn off the hot I/O detection for a device or devices for any length of time. Hot I/O detection protects CP from broken hardware that floods the system with unsolicited interrupts. If you turn hot I/O detection off, you may experience performance degradation or a system abend.

**nnnnn**


tells CP how many unsolicited interrupts per second to accept for the specified device or devices before CP stops accepting input from the device or devices. The variable *nnnnn* is a decimal number from 1 to 62500.

**Usage Notes**

1. CP processes SET HOTIO commands in the order that you enter them. If you enter more than one SET HOTIO command for a specific device or devices, the last command you enter is the one that CP uses for that device or devices. For example, if you issued the following commands:

```
set hotio all 25
set hotio 2ff 50
set hotio switch 75
```

CP sets the hot I/O rate for all devices in the active I/O configuration to 25, then CP sets the hot I/O rate for device X'2FF' to 50, and lastly, CP sets the hot I/O rate for all special switching devices to 75. If device X'2FF' is a special switching device, the third command overrides the second command and CP sets the hot I/O rate for device X'2FF' to 75, instead of 50.

2. If you want to set the hot I/O rate before the system IPLs, use the [HOT\\_IO\\_RATE Statement](#) in the system configuration file. For more information, see *z/VM: CP Planning and Administration*.
3. All real I/O devices have a hot I/O rate of 16 consecutive unsolicited interrupts per second, unless you override the default using the HOT\_IO\_RATE statement or the SET HOTIO command.
4.  **Attention:** We do not recommend that you specify the OFF operand to turn off the hot I/O detection for a device or devices for any length of time. Hot I/O detection protects CP from broken hardware that floods the system with unsolicited interrupts. If you turn hot I/O detection off, you may experience performance degradation or a system abend.

## SET HOTIO

### Examples

Example 1:

To change the hot I/O rate for all DASD devices in the active I/O configuration to 100 and to change the hot I/O rate for all devices in the range X'200' to X'300' in the active I/O configuration to 50, use the following SET HOTIO commands:

```
set hotio dasd 100
set hotio 200-300 50
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP6704E Missing token at end of line
- HCP6706E Invalid {number|string} - {*number*|*string*}

## SET IMSG



### Authorization

Privilege Class: G

### Purpose

Use SET IMSG to control the display of certain informational responses. The affected CP commands are:

- CHANGE
- CLOSE
- DEFINE
- DETACH
- ORDER
- PURGE
- REDEFINE
- SPOOL
- SPXTAPE
- TRANSFER

**Note:** The affected response is identified in each command description.

### Operands

#### ON

specifies that the responses are displayed.

#### OFF

specifies that the responses are not displayed.

#### IUCV

specifies that informational messages are to be passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the message is handled as if SET IMSG ON had been issued.

### Usage Notes

1. The SET IMSG ON or OFF command has no effect on the handling of error messages set by the SET EMSG command. The initial setting for IMSG is installation dependent and is defined in the system configuration file.
2. DETACH RDEV will still display informational messages when IMSG is set OFF. The other DETACH commands will not display informational messages when IMSG is set OFF.

### Messages

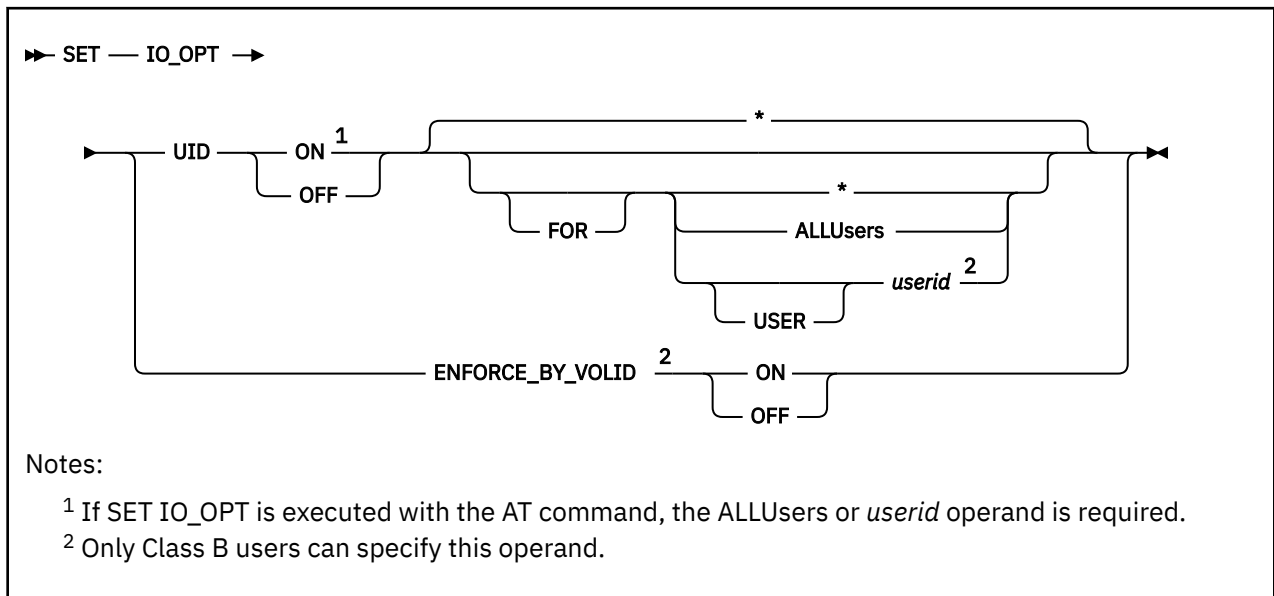
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

## **SET IMSG**

- HCP026E Operand missing or invalid



## SET IO\_OPT



### Authorization

Privilege Class: B, G

### Purpose

Use SET IO\_OPT to control whether or not virtual machine I/O options will be used for a virtual machine.

### Operands

#### UID

indicates guest PCI UID uniqueness enforcement will be affected.

#### ENFORCE\_BY\_VOLID

indicates the enforcing of attachment of DASD devices by VOLID will be affected.

#### ON

#### OFF

turns I/O options on or off for a virtual machine or system setting.

#### \*

asterisk represents the default. The default is that the command affects the virtual machine (User ID) only from where the command is issued.

#### ALLUsers

enables or disables I/O options for all virtual machines, independent of the individual settings for each user. The individual settings are not changed. When ALLUSERS is specified by a class B user, the specified I/O option is allowed or disallowed for all virtual machines, including any new users that might log on. When this global setting is OFF, it overrides any individual user's setting to enable the I/O option.

If SET IO\_OPT is executed with the AT command, the ALLUSERS or *userid* operand is required.

#### USER *userid*

specifies an individual user whose I/O option settings will be changed. If SET IO\_OPT is executed with the AT command, the ALLUSERS or *userid* operand is required.

### Usage Notes

1. If an IO\_OPT is set ON for ALLUSERS, an individual user can set that option OFF without changing the setting for other users.  
**Note:** The reverse is not true; setting OFF for ALLUSERS overrides individual user settings.
2. The QUERY IO\_OPT command can be used to see the current option settings for a given virtual machine.
3. The default for the UID option for the all users and each virtual machine is to be set ON.
4. No checks or changes to PCI functions already attached to virtual machines are made when the UID option is changed. Functions may need to be detached from virtual machines when the UID option is turned on to remove functions with identical UID values.
5. The ENFORCE\_BY\_VOLID option only affects the system wide setting which is initially determined by the ENFORCE\_BY\_VOLID configuration statement. See *z/VM: CP Planning and Administration* for more information on the [ENFORCE\\_BY\\_VOLID Statement](#) and its affects.

### Examples

Example 1:

The following command will disable userid test1 from using virtual machine UID I/O option.

```
set io_opt uid off test1
```

### Responses

Response 1:

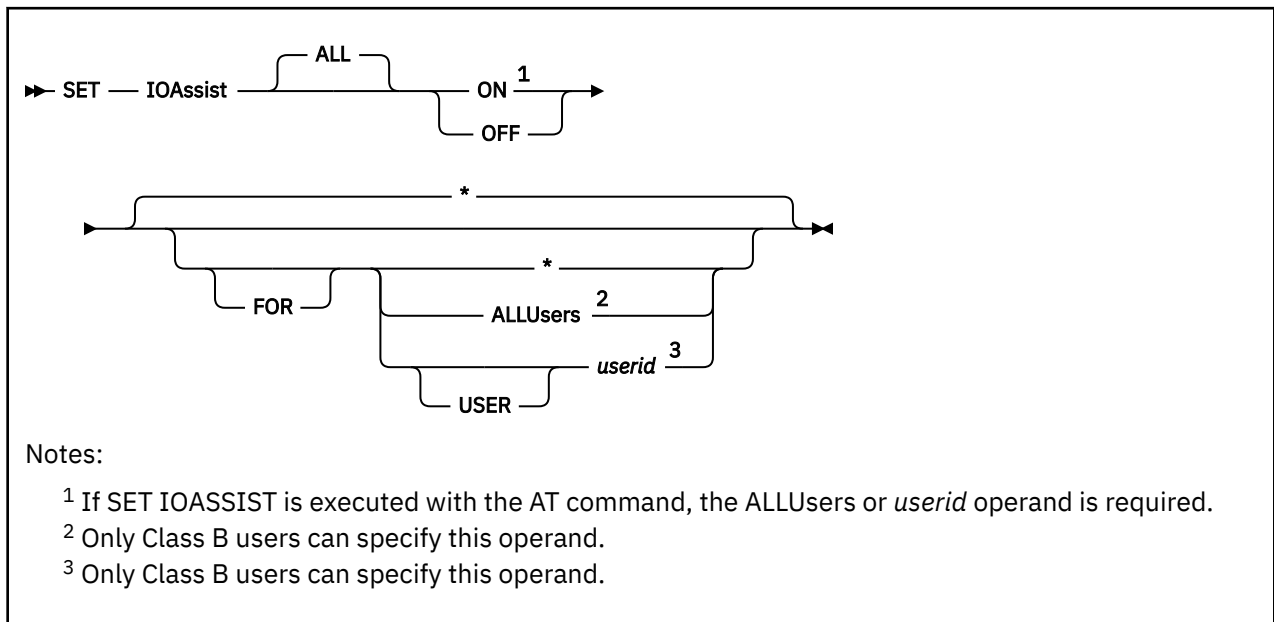
```
IO_OPT is already {enabled }  
                  {disabled}
```

indicates that the IO\_OPT specified is already enabled or disabled.

### Messages

- HCP002E Invalid operand - *operand*
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command
- HCP1014E A required option is missing

## SET IOASSIST



### Authorization

Privilege Class: B, G

### Purpose

Use SET IOASSIST to control whether or not virtual machine I/O assists will be used for a virtual machine.

To use any virtual machine I/O assists, the following are required:

- z/VM must be running in a logical partition and not as a guest.
- I/O assists must be globally enabled with the CP SET IOASSIST ON FOR ALLUSERS command.
- The virtual machine must be enabled with the CP SET IOASSIST ON FOR USER command.
- The virtual machine must not be using the CP TRACE command.
- Only the I/O assists that the processor supports will be enabled.

### Operands

#### ALL

indicates all I/O assists will be affected.

#### ON

#### OFF

turns I/O assists on or off for a virtual machine.

#### \*

asterisk represents the default. The default is that the command affects the virtual machine (User ID) only from where the command is issued.

#### ALLUsers

enables or disables eligibility for all virtual machines to use I/O assists, independent of the individual settings for each user. The individual settings are not changed. When ALLUSERS is specified by a class B user, access to I/O assists are allowed or disallowed for all virtual machines, including any new users that might log on. When this global setting is OFF, it overrides any individual user's setting to allow I/O assists.

## SET IOASSIST

If SET IOASSIST is executed with the AT command, the ALLUSERS or *userid* operand is required.

### USER *userid*

specifies an individual user whose assist settings will be changed. If SET IOASSIST is executed with the AT command, the ALLUSERS or *userid* operand is required.

## Usage Notes

1. If IOASSIST is set ON for ALLUSERS, an individual user can set IOASSIST OFF without changing the setting for other users.

**Note:** The reverse is not true; setting OFF for ALLUSERS overrides individual user settings.

2. Setting IOASSIST ON or OFF for all users may not immediately enable or disable the assist for all users. The command will take effect as soon as possible. Use the QUERY IOASSIST to determine when the user's virtual machine is in the desired state.

3. The SET QIOASSIST command and the SET IOASSIST command both disable all virtual machine I/O assists. Setting QIOASSIST ON/OFF for a virtual machine also performs the SET IOASSIST ALL ON/OFF command for the target virtual machine.

## Examples

Example 1:

The following command will disable *userid* test1 from using virtual machine I/O assists.

```
set ioassist off test1
```

## Responses

Response 1:

```
IOASSIST is already {enabled }  
                   {disabled}
```

indicates that the IOASSIST is now already enabled or disabled.

## Messages

- HCP002E Invalid operand - *operand*
- HCP007E Invalid *userid* - *userid*
- HCP020E *userid* missing or invalid
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command
- HCP1014E A required option is missing
- HCP2162I IOASSIST is not available

## SET IOCDS\_ACTIVE

►► Set — IOCDS\_active —<sup>1</sup>— An ◄◄

Notes:

<sup>1</sup> You must specify one IOCDS file in the range A0 through A3.

### Authorization

Privilege Class: B

### Purpose

Use SET IOCDS\_ACTIVE to change the input/output configuration data set (IOCDS) file that the processor will use during the next power-on reset (POR).

### Operands

**An**

is the file type of the IOCDS file that you want the processor to use during the next POR. The variable *n* is a decimal number between 0 and 3.

### Usage Notes

1. Once you issue the SET IOCDS\_ACTIVE command to change the active IOCDS file, the SE automatically puts the new IOCDS file in write-protect mode for you. This means that no one can make changes to the active IOCDS file.
2. If the current hardware configuration does not match the IOCDS file that you specify, CP issues error message HCP6814E and terminates command processing.
3. Whenever you issue one of the following dynamic I/O commands:

DEFINE CHPID	DEFINE PATH
DELETE CHPID	DELETE PATH
MODIFY CHPID	MODIFY PATH
DEFINE CU	DEFINE CNTLUNIT
DELETE CU	DELETE CNTLUNIT
MODIFY CU	MODIFY CNTLUNIT
DEFINE DEVICE	DEFINE IODEVICE
DELETE DEVICE	DELETE IODEVICE
MODIFY DEVICE	MODIFY IODEVICE

CP puts you in configuration mode, processes the dynamic I/O command, and then takes you out of configuration mode.

4. Before you can use any I/O devices, you must include those devices in the hardware's and software's view of the I/O configuration.

The software creates its view of the I/O configuration at initialization either by sensing the I/O device, by reading RDEVICE statements in the system configuration file, or some combination thereof. After initialization, you can change the software's view of the I/O configuration using the SET RDEVICE command (see [“SET RDEVICE” on page 1627](#)) or the dynamic I/O commands listed in Usage Note [“3” on page 1525](#).

The hardware creates its view of the I/O configuration at power-on reset (POR) time. The I/O configuration is defined in the I/O configuration program (IOCP) source file that is stored on a user's

## SET IOCDS\_ACTIVE

minidisk (or SFS directory). You run the IOCP utility against the IOCP source file to create an I/O configuration data set (IOCDS) file that is stored on the support element (SE) hard disk. You can have multiple IOCDS files stored on the SE hard disk. At power-on reset time, you choose the IOCDS file that you want to use. The specified IOCDS file is copied to the hardware system area (HSA) storage and becomes the active hardware I/O configuration. Thus, there are 2 parts to the hardware's view of the I/O configuration: the active IOCDS file in HSA storage and the IOCDS file on the SE hard disk.

After the machine is up and running, you can change the hardware's view of the active I/O configuration using the dynamic I/O commands listed in Usage Note “3” on page 1525. However, these commands only change the active I/O configuration in the HSA. They do not change the IOCDS file on the SE hard disk.

To change the IOCDS file on the SE hard disk, you must:

- a. Update the IOCP source file.
- b. Run the IOCP utility (see “IOCP” on page 2203) against the updated IOCP file. Specify all options you normally specify including WRTxx (to write the newly generated IOCDS file to the SE hard disk). If the IOCP utility runs without errors, it downloads the new IOCDS file to the SE hard disk.
- c. Issue the SET IOCDS\_ACTIVE command to make the new IOCDS file the active file that the processor will use during the next power-on reset (POR).

## Responses

Response 1:

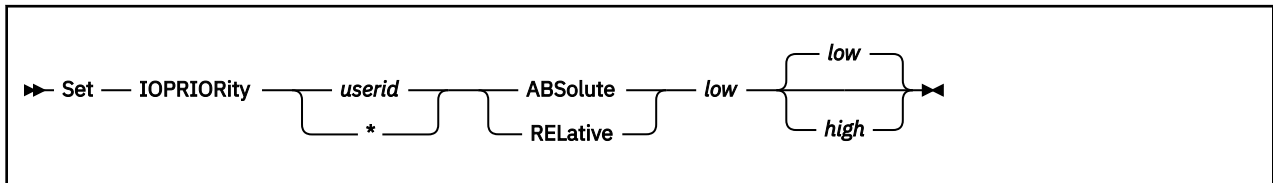
To change the active IOCDS file to A3, enter the following:

```
set iocds_active a3
The active IOCDS file is now A3
Ready;
```

## Messages

- HCP026E Operand missing or invalid
- HCP6812E Set IOCDS\_active command failed with return code *nnnn*
- HCP6814E Command is not valid on this processor

## SET IOPRIORITY



### Authorization

Privilege Class: A

### Purpose

Use SET IOPRIORITY to change a user's I/O priority settings.

### Operands

#### *userid*

\*

is the user identification of the virtual machine whose I/O priority range changed. That virtual machine must be logged on to the system. You can specify your user ID as an asterisk (\*).

#### **ABSolute**

indicates the type of I/O priority range. If LPAR I/O Priority Facility is enabled, the I/O priority range specified must fit onto the range available to CP. If the range specified falls outside of the range available to CP, an informational message is displayed and the requested range is clipped to fall within the range available to CP.

If LPAR I/O Priority Facility is not enabled or not installed, CP simulates an I/O priority range from 0 to 255.

#### **RELative**

indicates the type of I/O priority range. The maximum I/O priority range is 0 to 255. If LPAR I/O Priority Facility is installed and enabled, the maximum I/O priority range maps proportionately onto the range available to CP with the highest I/O priority mapping directly to the highest value available to CP. The requested RELATIVE I/O priority range is used to calculate a percentage of the maximum I/O priority range, and the user's effective I/O priority range is the corresponding percentage of the I/O priority range available to CP.

The following formula is used to calculate an effective I/O value:

$$\text{Eff\_Val} = \text{TRUNC}\left(\frac{(\text{Rel\_Val} - \text{Rel\_Lo}) * (\text{CP\_Hi} - \text{CP\_Lo})}{255}\right) + \text{CP\_Lo}$$

where

<b><u>Value</u></b>	<b><u>Meaning</u></b>
Eff_Val	Effective I/O priority
Rel_Val	Relative I/O priority
Rel_lo	Relative low priority
CP_Hi	High I/O priority available to CP
CP_Lo	Low I/O priority available to CP

## SET IOPRIORITY

If LPAR I/O Priority Facility is not enabled or not installed, CP simulates an I/O priority range from 0 to 255. Thus, the maximum I/O priority range maps directly onto the range simulated by CP.

### **low**

is the low value of the I/O priority range. It must be a number from 0 to 255.

### **high**

is the high value of the I/O priority range. It must be a number from 0 to 255 and greater than or equal to the low value. If not specified, the high value is equal to the low value.

## Usage Notes

1. The initial I/O priority settings for a user can be specified by the IOPRIORITY statement in the user's directory entry. If an IOPRIORITY statement is not found and LPAR I/O Priority Facility is enabled, LOGON processing requests RELATIVE I/O priority range with a low value of 0 and a high value of 0. The user's effective range is computed based on the I/O priority range available to CP.
2. The I/O priority settings of a user can be displayed by the QUERY IOPRIORITY command. For more information on this command, see [“QUERY IOPRIORITY”](#) on page 952.

## Responses

Response 1:

To set I/O priority of user USER1 to an absolute range with a low value of 3 and a high value of 3, enter the following:

```
set iopriority user1 abs 3
Command complete
```

## Examples

1. Suppose a user requests a RELATIVE I/O priority range of 40 to 100, and the I/O priority range available to CP is 0 to 15. For relative ranged, the maximum I/O priority range (0–255) maps proportionately onto the range available to CP with the highest I/O priority mapping directly to the highest value available to CP. The following table shows how the entire range of relative I/O priorities map onto the range available to CP:

Relative Value	Maps to CP's Priority
0 to 16	0
17 to 33	1
34 to 50	2
51 to 67	3
68 to 84	4
85 to 101	5
102 to 118	6
119 to 135	7
136 to 152	8
153 to 169	9
170 to 186	10
187 to 203	11
204 to 220	12



Relative Value	Maps to CP's Priority
221 to 237	13
238 to 254	14
255	15

Therefore, a user requesting relative I/O priority range of 40 to 100 would have effective I/O priority range of 2 to 5.

2. As in the previous example, suppose a user requests a relative I/O priority range of 40 to 100 but this time the I/O priority range available to CP is 8 to 14. The following table shows how the entire range of relative I/O priorities map onto the range available to CP:

Relative Value	Maps to CP's Priority
0 to 42	8
43 to 84	9
85 to 127	10
128 to 169	11
170 to 212	12
213 to 254	13
255	14

A user requesting relative I/O priority range of 40 to 100 would have effective I/O priority range of 8 to 10.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*}
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP2092E Second value of range must be greater than or equal to value of range
- HCP2094I Requested ABSOLUTE I/O priority range falls outside range currently allowed by CP and has been changed to fit within CP's range
- HCP2768E Missing I/O priority
- HCP6704E Missing token at end of line
- HCP6706E Invalid I/O priority - *priority*

## SET IPLPARMS

►► Set — IPLparms ◄◄

### Authorization

Privilege Class: A

### Purpose

Use SET IPLPARMS to change the IPL parameters currently in CP storage that CP will use for a shutdown re-IPL or a software restart.

### Usage Notes

1. The SET IPLPARMS command prompts the user for up to 3 lines of data. Each line is either truncated to 80 characters or padded with blanks out to 80 characters. These lines are joined together with no intervening blanks to form the IPL parameters.
2. Specify all of the IPL parameters you want CP to use, even if they are already in CP storage. The SET IPLPARMS command replaces all the IPL parameters in CP storage. For example, if you IPLed with the IPL parameter to override the default console address (CONS) and then issued a SET IPLPARMS command to override the default file name of the system configuration file (FN), CP storage would contain only the new value for FN. The next shutdown re-IPL or software restart would use the default console address because you did not specify it on the SET IPLPARMS command.
3. The IPL parameters (listed below) are described in *z/VM: System Operation*.

Parameter	Purpose
CLEARPDR	Reinitialize the PDR when all other SSI members are down.
CONS= or CON=	The operator console address.
FN=	The system configuration file name.
FT=	The system configuration file type.
IPLVAR=	The value of the CP.IPLPARMS.IPLVAR system environment variable.
NOEXITS	Do not pass certain options in the configuration file.
NOHCD	Ignore the IODF statement in the system configuration file.
PDNUM=	The parm disk number.
PDOFF=	The parm disk offset in the IPL volume.
PDVOL=	The real address of the parm disk.
PROMPT	The prompt for startup options at IPL.
REPAIR	Bypass SSI cluster initialization and management functions. For non-SSI, this is equivalent to specifying the NOAUTOLOG and DISABLE IPL options.
<i>selector:value (IPL parameter substring)</i>	Define conditional IPL parameters based on the value of the specified <i>selector</i> .
STORE=	The storage amount.

4. After all of the IPL parameter data has been accepted, CP checks each parameter for syntax errors. If you make a syntax error (typing the name of an IPL parameter incorrectly, specifying a non-numeric value for an IPL parameter that is expecting a number, and so forth), CP issues error message HCP6770E and ignores your SET IPLPARMS command. CP does not change the IPL parameters in CP storage unless the SET IPLPARMS completes without errors. See **Response 3** for an example of making a typographical error in the name of an IPL parameter.
5. To display the IPL parameters in CP storage, use the QUERY IPLPARMS command (see [“QUERY IPLPARMS”](#) on page 954).

## Responses

### Response 1:

To set or reset the IPL parameters that CP will use after the next SHUTDOWN REIPL command or CP abend, enter the following SET IPLPARMS command:

```
set iplparms
ENTER UP TO 3 LINES. ENTER A NULL LINE TO END.
FN=SYSTEM FT=CONFIG CONS=08E6
PROMPT

IPL parameters have been replaced.
Ready;
```

### Response 2:

To clear the IPL parameters in CP storage so that CP will use the default values, use the following SET IPLPARMS command:

```
set iplparms
ENTER UP TO 3 LINES. ENTER A NULL LINE TO END.

IPL parameters have been replaced.
Ready;
```

### Response 3:

If you enter a mistake after the prompt, CP does not replace any of the IPL parameters, even the ones that were not in error. For example, suppose you wanted to set (or reset) the IPL parameters as in **Response 1**, but you made a typographical error. CP would ignore your entire SET IPLPARMS command, not just the IPL parameter in error:

```
set iplparms
ENTER UP TO 3 LINES. ENTER A NULL LINE TO END.
FN=SYSTEM FT=CONFIG CONS=08E6
PRIMPT

HCP6770E IPL parameters: Unrecognized option - PRIMPT
IPL parameters have not been replaced.
Ready(06770);
```

### Response 4:

If you enter an IPL parameter more than once, CP stores the last occurrence in CP storage and ignores all previous occurrences. For example, suppose you wanted to set (or reset) the IPL parameters as in **Response 1**, but you specified the wrong file name for the system configuration file. If you catch the mistake before you enter the null line that ends the SET IPLPARMS command, you can enter another file name:

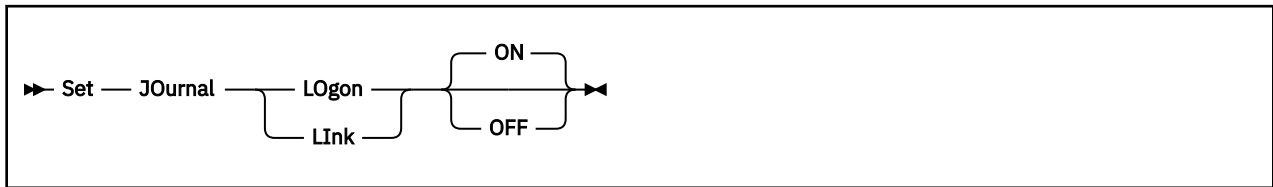
```
set iplparms
ENTER UP TO 3 LINES. ENTER A NULL LINE TO END.
FN=SYSTM FT=CONFIG CONS=08E6
PROMPT
FN=SYSTEM
```

IPL parameters have been replaced.  
Ready;

### Messages

- HCP003E Invalid option - {command|statement} contains extra option(s) starting with *option*
- HCP2872E Command line is too long for CP. CP can accept no more than 240 characters. Your command was *nnn* characters in length
- HCP6014I *command* cancelled at user request.
- HCP6015E *command* cancelled because of console read error.
- HCP6770E IPL parameters: Expected ( not found following *selector*
- HCP6770E IPL parameters: Expected ) but found end of string
- HCP6770E IPL parameters: Invalid console address - *address*
- HCP6770E IPL parameters: Invalid file name - *fn*
- HCP6770E IPL parameters: Invalid file type - *ft*
- HCP6770E IPL parameters: Invalid parm disk number - *n*
- HCP6770E IPL parameters: Invalid parm disk offset - *offset*
- HCP6770E IPL parameters: Invalid parm disk volume - *address*
- HCP6770E IPL parameters: Invalid selector value - *value*
- HCP6770E IPL parameters: Invalid storage specification - *nnnn*
- HCP6770E IPL parameters: Invalid IPLVAR value - *value*
- HCP6770E IPL parameters: No value specified for selector *selector*
- HCP6770E IPL parameters: Option *option* ignored. No longer supported
- HCP6770E IPL parameters: Selector not allowed - *selector*
- HCP6770E IPL parameters: Unrecognized option - *option*
- HCP6770E IPL parameters: Unrecognized selector - *selector*

## SET JOURNAL



### Authorization

Privilege Class: A

### Purpose

Use SET JOURNAL to start or stop the journal security function.

### Operands

#### LOGON ON

#### LOGON OFF

indicates whether the LOGON, AUTOLOG, and XAUTOLOG thresholds established in the system configuration file at system generation are observed. The default is ON.

#### LINK ON

#### LINK OFF

indicates whether type 05, 06 and 08 accounting records are generated and messages are sent to operations personnel when the LINK thresholds established in the system configuration file at system generation are violated. The default is ON.

### Usage Notes

1. The SET JOURNAL command can be used only if the FACILITY ON operand and SET\_AND\_QUERY ON operand are specified on the JOURNALING statement in the system configuration file.
2. SET JOURNAL LINK OFF changes to 10 the limit for incorrect password LINK attempts that a particular user can enter.
3. SET JOURNAL LOGON OFF disables the logon inductor and discards the count of logon attempts associated with terminals and user IDs that are not logged on at the time. If SET JOURNAL LOGON ON is then issued, no information about previous logon attempts is available.

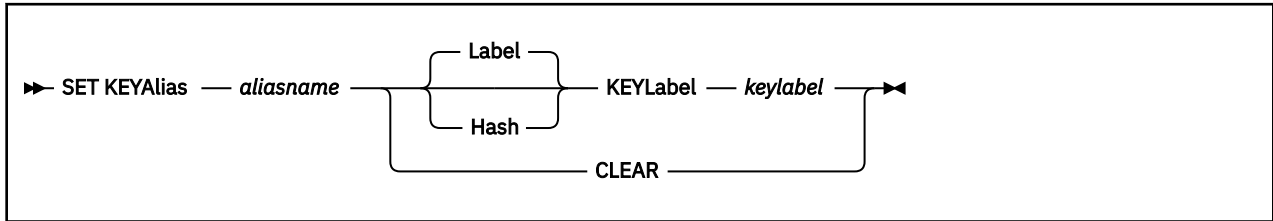
**Note:** The logon inductor is the CP facility that responds to excessive logon attempts with a bad password. It temporarily prohibits a terminal or user ID from logging on.

4. Users who exceed the AUTOLOG/XAUTOLOG not-valid-password limit and are not permitted to use these commands for the remainder of their logon session, have their ability to use these commands restored after SET JOURNAL LOGON OFF is issued. If SET JOURNAL LOGON ON is issued later, these users again are unable to enter AUTOLOG or XAUTOLOG. Refer to Usage Note “5” on page 1432 in SET CMDLIMIT for details on what will happen when SET JOURNAL LOGON OFF is issued for invalid password attempts.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid

## SET KEYALIAS



### Authorization

Privilege Class: B

### Purpose

Use SET KEYALIAS to associate a shorter or more meaningful alias name with a key label. Key label aliases are used on the ATTACH and SET RDEV commands to identify the key label(s) to be used for encrypting data written to a tape.

### Operands

#### *aliasname*

a 1 to 32 character string alias name. Imbedded blanks and lowercase characters are permitted. Enclose the string in single quotation marks to include blanks in the alias name. Alias names are not case sensitive.

#### **Label**

indicates the key label encoding mechanism. The key label is to be used as is. This is the default.

#### **Hash**

indicates the key label encoding mechanism is a hashed value of the key label.

#### *keylabel*

a 1 to 64 character key label known to the key manager being used for tape encryption. Imbedded blanks and lowercase characters are permitted. Enclose the string in single quotation marks to include blanks in the key label. Key labels are not case sensitive.

#### **CLEAR**

indicates the existing *aliasname* definition is to be deleted.

### Usage Notes

1. If the alias name being defined already exists it will be replaced by the new definition.
2. Use the CLEAR option to delete a key alias definition.

### Responses

Response 1:

To define an alias called "myalias" which does not contain imbedded blanks and uses the default encoding mechanism, enter:

```
set keya myalias key1 e3rw33rssesyqypsftqpx0539
Ready;
```

Response 2:

To define an alias called "THIS IS MY ALIAS" which contains imbedded blanks and uses HASH encoding, enter:

```
set keya 'this is my alias' hash keyl e3rw33rssesyqypsftqpx0539
Ready;
```

Response 3:

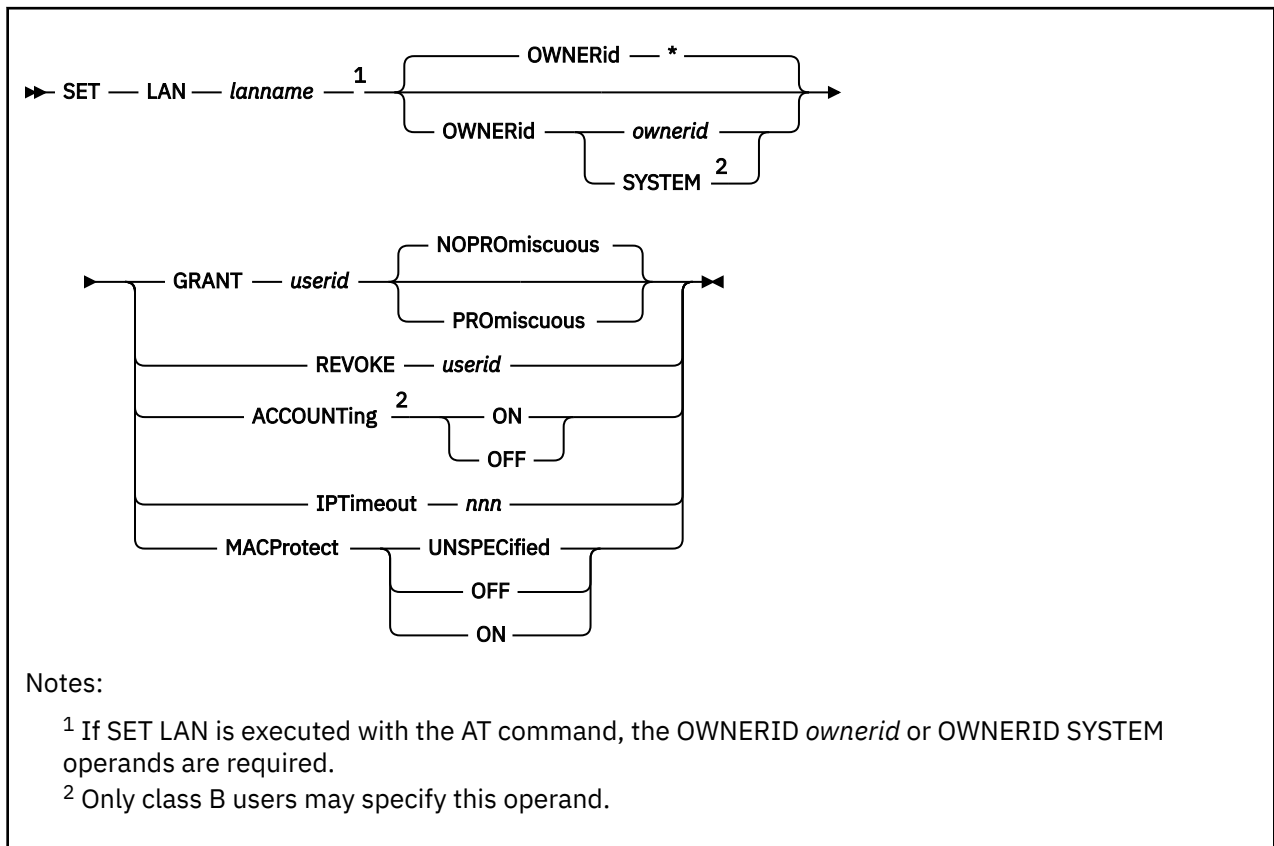
Although key aliases and labels are case insensitive, it is possible to assign mixed cased values. An alias containing lowercase characters can be defined using a REXX EXEC as shown below. Note that the CP commands within the command string being executed must be uppercase.

```
(a simple REXX exec)
/* This comment format indicates the REXX interpreter is to be used */
address command "CP SET KEYA 'ABC Co' KEYL e3rw33rssesyqypsftqpx05399"
```

## Messages

- HCP002E Invalid operand - *operand*.
- HCP003E Invalid option - command contains extra option(s) starting with *option*.
- HCP263E Too many operands were supplied for this command.
- HCP6704E Missing token at end of line.
- HCP6706E Invalid string - *string*.
- HCP9968E Key alias not found: *alias*.
- HCP9968E No key aliases found.
- HCP9968E The key alias contains leading or trailing blanks or an empty string.
- HCP9968E The key label contains leading or trailing blanks, invalid characters, or an empty string.

## SET LAN



### Authorization

Privilege Class: B, G

### Purpose

Use SET LAN to modify the attributes of a guest LAN. The current owner of the LAN (or a Class B user) can:

- Grant permission for a user to connect (COUPLE)
- Revoke permission for a user to connect (COUPLE)
- Modify the IPTimeout value for the guest LAN
- Modify the MAC protection setting for the guest LAN

The Class B form of the command allows the invoker to modify a LAN owned by another user. Only a Class B user can specify OWNERid SYSTEM (modifying a SYSTEM LAN). Only a Class B user can specify ACCOUNTING to turn accounting on or off for a LAN.

### Operands

#### *lanname*

is the name of the guest LAN to be modified. Both *lanname* and *ownerid* must be known to fully qualify a LAN.

#### OWNERid \*

#### OWNERid *ownerid*

#### OWNERid SYSTEM

identifies the owner of the subject LAN. When OWNERID \* is specified, it indicates the invoker is the owner of the LAN. This is the default when OWNERID is omitted.



The Class G form of the command only accepts \* or the user ID of the invoker.

The Class B form of the command accepts another user ID (like OWNERID SYSTEM).

#### **IPTimeout *nnn***

indicates the length of time in minutes that a remote IP address table entry remains in the IP address table for the guest LAN.

*nnn* is a number from 1 to 240.

#### **GRANT *userid***

specifies *userid* is added to the access list for this LAN. This allows user *userid* to connect an adapter to LAN *lanname* (if it is a RESTRICTED LAN). If an External Security Manager (ESM) is in control of the LAN, it may override the CP access list.

#### **PROMiscuous**

specifies that the *userid* from the grant is authorized for PROMISCUOUS Mode.

#### **NOPROMiscuous**

specifies that the *userid* from the grant is NOT authorized for PROMISCUOUS Mode.

If an External Security Manager (ESM) is in control of the LAN, it may override the PROMISCUOUS authority.

#### **REVOKE *userid***

specifies *userid* is removed from the access list for this LAN. This prevents user *userid* from connecting an adapter to a RESTRICTED LAN *lanname*. If an External Security Manager (ESM) is in control of the LAN, it may override the CP access list. Existing connections to the LAN are not affected.

#### **ACCOUNTing ON**

#### **ACCOUNTing OFF**

allows a Class B user to control whether accounting records are created for the LAN specified.

#### **MACProtect**

modifies the MAC address protection for devices coupled to this guest LAN. Turning MAC address protection on prevents a guest from using a MAC address that is not assigned to the user's network device. MACPROTECT is valid only for an ETHERNET guest LAN.

There are three levels of inheritance used when determining the MAC address protection level for a MAC address assigned to a network device. The highest is the system level, followed by the guest LAN (or virtual switch) level. The lowest level is the protection set for a specific network data device.

The MAC address protection level assigned is determined first at the device, then guest LAN and lastly the system level. If MAC address protection is set at the device level using the SET NIC CP command, then its current setting is used. When the device is set to UNSPECIFIED, it will inherit the guest LAN's MACPROTECT setting. When the guest LAN MACPROTECT setting is set to UNSPECIFIED, the SYSTEM's MAC address protection specified by the SET VMLAN CP command or VMLAN configuration statement is assigned.

#### **UNSPECified**

allow the MAC address protection to be determined by the SYSTEM level setting. UNSPECIFIED is the default setting for a guest LAN.

If MAC address protection is set to UNSPECIFIED by the SET NIC CP command for a specific device, then the device's MAC address protection will be set to the SYSTEM level defined by the SET VMLAN CP command or the VMLAN configuration statement.

#### **OFF**

set the MAC address protection level off for coupled devices defined with a MAC address protection set to UNSPECIFIED.

If MAC address protection is set to UNSPECIFIED by the SET NIC CP command for a specific device, then the device's MAC address protection will be set off.

#### **ON**

set the MAC address protection level on for coupled devices defined with a MAC address protection set to UNSPECIFIED.

## SET LAN

If MAC address protection is set to UNSPECIFIED by the SET NIC CP command for a specific device, then the device's MAC address protection will be set on.

### Usage Notes

1. If the guest LAN is defined with the UNRESTRICTED attribute, any user can connect a virtual adapter to that LAN. The CP access list has no meaning for an UNRESTRICTED LAN. This will be overridden if an External Security Manager (ESM) is in control of the guest LAN.

### Examples

The following examples apply when CP, not an ESM, is protecting the guest LAN:

1. SET LAN - Class G:

To modify a user LAN named TECHINFO by adding user REYNOLDS to the access list, enter:

```
cp set lan techinfo grant reynolds
command complete
```

2. SET LAN - Class B:

To modify a SYSTEM LAN named ADMIN by adding user MAINT to the access list, enter:

```
cp set lan admin ownerid system grant maint
command complete
```

**Note:** This must be done by a user with Class B privileges since the LAN is not owned by the invoker.

3. SET LAN - Class B:

To begin generating account records for a SYSTEM LAN named ADMIN enter:

```
cp set lan BKT614 accounting on
command complete
```

**Note:** This must be done by a user with Class B privileges since the ACCOUNTING option is specified.

### Responses

Response 1:

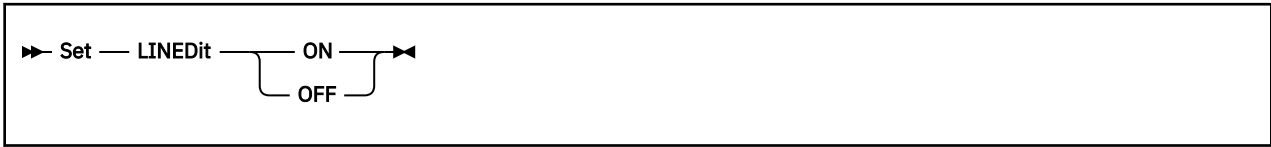
```
command complete
```

is the normal response from this command.

### Messages

- HCP2783E LAN *ownerid lanname* does not exist
- HCP2795E Promiscuous Mode can not be used with a HiperSockets device
- HCP2795E You are not authorized to override LAN ACCOUNTING
- HCP2795E You are not authorized to SET LAN *lanname* for OWNERID *ownerid*
- HCP2858I *userid* connection to *lanowner lanname* has been [revoked | updated] by *userid1*.
- HCP2858I *userid* connection to *lanowner lanname* is not affected by this update.
- HCP3022E SET LAN not performed; MACPROTECT is not valid for the IP transport type.

## SET LINEDIT



### Authorization

Privilege Class: G

### Purpose

Use SET LINEDIT to control logical line edit functions.

### Operands

#### ON

#### OFF

controls the logical line editing functions for your virtual machine. If you specify ON, the system line editing functions and symbols are used for editing virtual console input requests. If you specify OFF, no character or line editing is used for the virtual machine operating system. When you log on, LINEDIT is set ON.

### Usage Notes

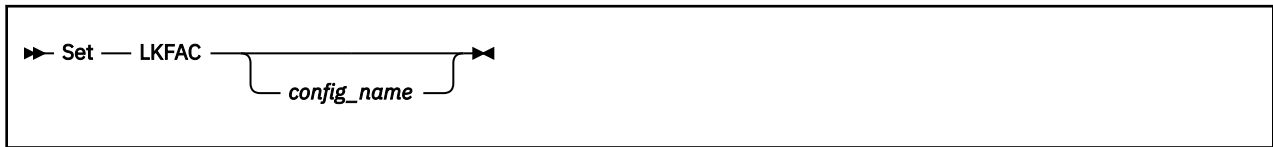
1. The character displayed depends on the code page used by the terminal emulator. The following characters are from code page 037 United States:

Character	Meaning
# X'7B'	SYSLEND LINEND
¢ X'4A'	SYSLDEL LINEDEL
@ X'7C'	SYSLCEL CHARDEL
" X'7F'	SYSLESCP ESCAPE
X'6A'	SYSTAB TABCHAR
[ X'AD'	SYSALDEL LINEDEL

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid

## SET LKFAC



### Authorization

Privilege Class: G

### Purpose

Use SET LKFAC to connect or disconnect a user to a specified Multi-Path Lock Facility RPQ (MPLF) simulation configuration.

Once connected, the user is identified as a member of that configuration. This means that the user's attached minidisk DASDs will be included in the specified MPLF I/O configuration and assigned a virtual MPLF DASD controller subsystem identifier (SSID). While a user is connected to a MPLF configuration, all MPLF locking functions are simulated for the user's attached minidisk devices.

MPLF device simulation is based on the assumption of 32 device numbers per block or storage director range. The range shares a virtual subsystem identifier and common lock space.

### Operands

#### *config\_name*

is the 1- to 8-character alphanumeric name of the MPLF configuration to be connected to or disconnected from.

### Usage Notes

1. If you specify *config\_name*, CP searches for that name in the configuration name table which contains currently defined MPLF configurations. If the name is found, the user is connected to the configuration. If the name is not found, it is added to the table and the user is then connected to the configuration.
2. If you enter SET LKFAC without *config\_name*, the user is disconnected from the MPLF configuration. If the user is the last connected user, the MPLF configuration is deleted.
3. There is a maximum of 25 configurations and a maximum of 32 userids assigned to each configuration.

### Responses

Response 1:

```
LKFAC CONFIGURATION config_name CREATED
```

is issued if this is the first time a user is connecting to the configuration.

#### *config\_name*

is the name of the configuration specified on the SET LKFAC command.

Response 2:

```
userid ADDED TO LKFAC CONFIGURATION config_name
```

tells you that the specified configuration already exists and connects the user to the configuration.

#### *userid*

is the user ID of the user issuing the SET LKFAC command.

***config\_name***

is the name of the configuration specified on the SET LKFAC command.

Response 3:

```
userid REMOVED FROM LKFAC CONFIGURATION config_name
```

is issued if the SET LKFAC command is issued without parameters; the user is disconnected from the configuration previously connected to.

***userid***

is the user ID of the user issuing the SET LKFAC command.

***config\_name***

is the name of the configuration from which the user was disconnected.

Response 4:

```
LKFAC CONFIGURATION config_name DELETED
```

is issued when the last user is disconnected from the specified configuration.

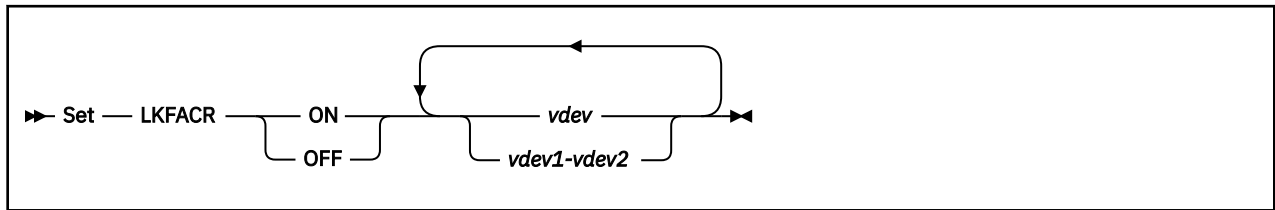
***config\_name***

is the name of the configuration specified on the SET LKFAC command.

**Messages**

- HCP9951E Invalid LKFAC configuration name entered
- HCP9952E Maximum (32) users assigned to LKFAC configuration *config\_name*
- HCP9954E Maximum LKFAC configurations defined, *config\_name* not created
- HCP9955E No change,LKFAC configuration same as entered
- HCP9964E User not authorized to use LKFAC simulation support

## SET LKFACR



### Authorization

Privilege Class: G

### Purpose

Use SET LKFACR to control whether a full-pack minidisk is to use the real multi-path lock facility (MPLF) of an IBM DASD storage controller.

### Operands

#### ON

indicates that the device will use the real MPLF facility

#### OFF


indicates that the device will not use the real MPLF facility.

#### vdev

#### vdev1-vdev2

is the virtual device number or range of virtual device numbers of the devices to be enabled or disabled for real MPLF.

### Usage Notes

- 
**Attention:** SET LKFACR ON causes a SYSTEM RESET of your virtual machine whenever one or more of the specified virtual devices have been enabled for MPLF simulation through the SET LKFACR command. Use the QUERY LKFACR command just before issuing SET LKFACR ON to see a current list of the virtual devices that would cause such a SYSTEM RESET of your virtual machine.
- SET LKFACR ON will prevent the virtual device from using simulated MPLF until the next SYSTEM RESET.
- The virtual machine must have lock facility authorization in order to use the real MPLF facility. See [OPTION Directory Statement \(LKFAC\)](#) in *z/VM: CP Planning and Administration*.
- SET LKFACR ON is only valid for full-pack minidisks. Non-full-pack minidisks can only use the simulated MPLF facility. For more information, see [“SET LKFAC” on page 1540](#). Dedicated DASD always use the real MPLF facility, provided that the virtual machine has the proper authorization. See [OPTION Directory Statement \(LKFAC\)](#) in *z/VM: CP Planning and Administration*.

### Responses

Response 1:

```
Real MPLF is {enabled } for DASD vdev
           {disabled}
```

is issued when the command completes successfully.

## Response 2:

```
Real MPLF is already enabled for DASD vdev - this device is
dedicated and always has access to the real MPLF.
```

is issued when the target device is dedicated.

## Response 3:

```
Real MPLF remains enabled for DASD vdev - this device is
dedicated and always has access to the real MPLF.
```

is issued when the target device is dedicated.

## Response 4:

```
Real MPLF remains disabled for DASD vdev - this device is
a non-full-pack minidisk and cannot access a real MPLF.
```

is issued when the target device is a non-full-pack minidisk.

## Response 5:

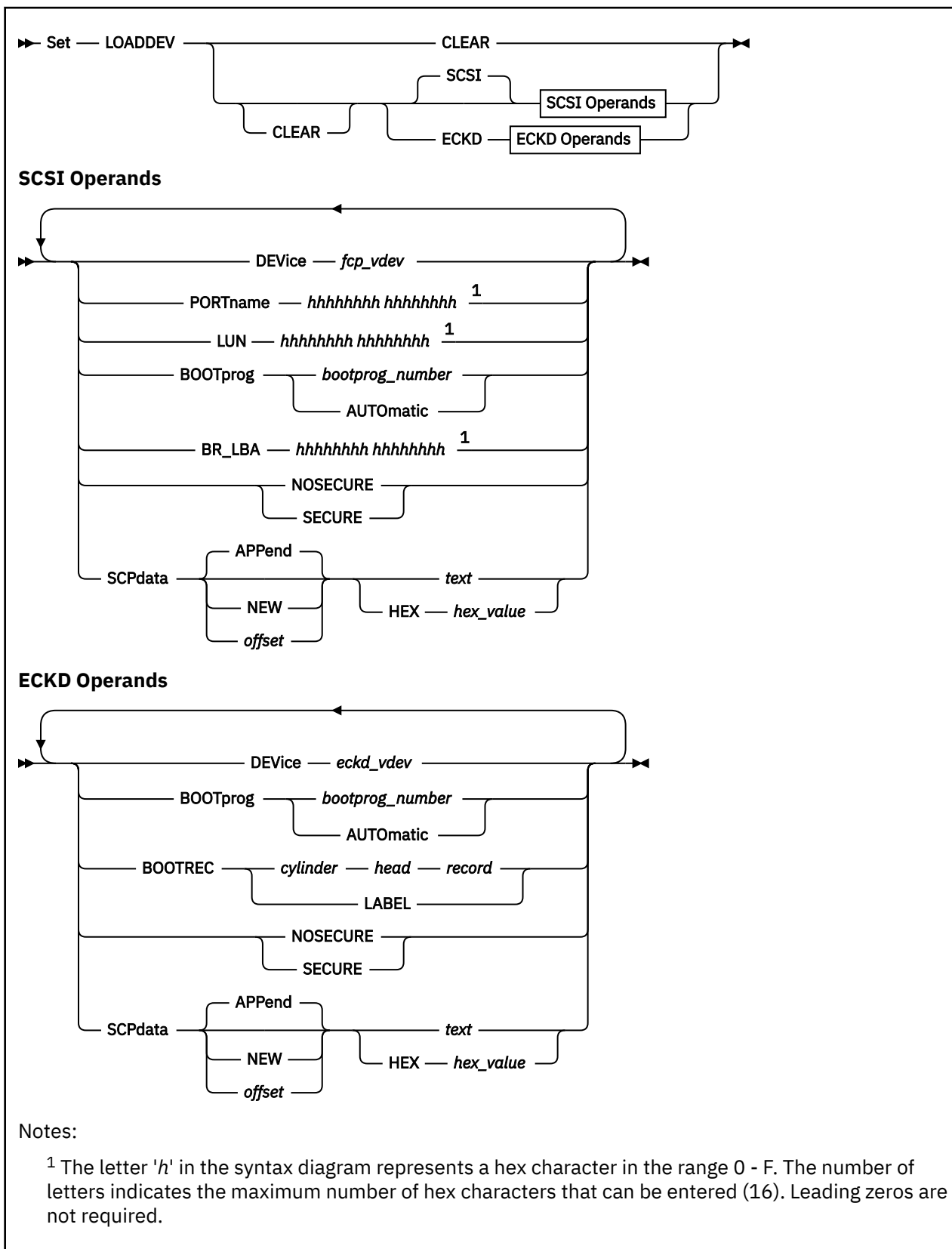
```
Real MPLF remains disabled for DASD vdev - Real MPLF
is not available when CHPID virtualization is active.
```

is issued when this command is invoked with CHPID virtualization on.

## Messages

- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP040E Device *{vdev}* does not exist
- HCP6706E Invalid address range - *range*
- HCP9966E You do not have LKFAC directory authorization
- HCP9967E *type vdev* LKFACR cannot be used - no real MPLF is available

## SET LOADDEV





## Authorization

Privilege Class: G

## Purpose

Use the **SET LOADDEV** command to set LOADDEV parameters for a guest IPL (without dump) from a SCSI or ECKD disk. The parameters can identify the virtual device number and specify whether the IPL is a secure IPL. The parameters can identify the location of an operating system program and data to pass to the program.

The parameters that are set by using the **SET LOADDEV** command are called LOADDEV parameters and are used only for list-directed IPLs, which occur in the following cases:

- The **IPL** command specifies the *fcp\_vdev* operand but not the DUMP operand.
- The **IPL** command specifies the LOADDEV operand.

See usage note [“1” on page 1547](#).

The settings are temporary. The settings persist after an IPL but not after the user logs off. LOADDEV statements in the user directory file are not changed. For more information, see [“IPL” on page 566](#).

## Operands

### CLEAR

The CLEAR operand clears all LOADDEV parameters for all device types (SCSI and ECKD). Operands that follow the CLEAR operand are not cleared.

After LOADDEV parameters are cleared, a list-directed IPL from SCSI or ECKD disk cannot be initiated until all required parameters for the SCSI or ECKD device type are set. The parameters that are required for each load device type are explained in usage note [“1” on page 1547](#).

### SCSI

The SCSI operand indicates that the LOADDEV parameters are for IPL of a SCSI device. Other operands on the command must be operands for a SCSI device.

If you do not specify the SCSI or ECKD operand, the device type defaults to SCSI.

After the SCSI parameter is set, subsequent **SET LOADDEV** commands cannot set ECKD parameters until a **SET LOADDEV** command with the CLEAR operand is specified. Likewise, LOADDEV parameters for an ECKD device must be cleared before you can set LOADDEV parameters for a SCSI device.

### ECKD

The ECKD operand indicates that the LOADDEV parameters are for IPL of an ECKD device. Other operands on the command must be operands for an ECKD device.

Because SCSI is the default device type, the ECKD operand must be specified on every command that sets LOADDEV parameters for an ECKD device.

After the ECKD parameter is set, subsequent **SET LOADDEV** commands cannot set SCSI parameters until a CLEAR operand is specified. Likewise, LOADDEV parameters for a SCSI device must be cleared before you can set LOADDEV parameters for an ECKD device.

### DEVICE *fcp\_vdev*

The DEVICE value (*fcp\_vdev*) specifies the virtual device number of the FCP device to IPL.

The LOADDEV DEVICE parameter must be set before you can issue the **IPL LOADDEV** command. A secure IPL without dump from a SCSI device can be initiated only by issuing the **IPL LOADDEV** command.

If the **IPL** command specifies the *fcp\_vdev* operand, then the LOADDEV DEVICE parameter must match that value or must not be set.

A table of restrictions and requirements for using the DEVICE parameter is presented in usage note [“1” on page 1547](#).

## SET LOADDEV

There is no default value for the DEVICE parameter.

### **DEVICE** *eckd\_vdev*

The DEVICE value (*eckd\_vdev*) specifies the virtual device number of the ECKD device to IPL.

The LOADDEV DEVICE parameter must be set before you can issue the **IPL LOADDEV** command. A secure guest IPL without dump from an ECKD device can be initiated only by issuing the **IPL LOADDEV** command.

### **PORTname** *hhhhhhhh hhhhhhhh*

The PORTNAME value (*hhhhhhhh hhhhhhhh*) specifies the 8-byte fibre channel port name of the SCSI device that is IPLed. The value must be a hexadecimal value in the range 0 - FFFFFFFF FFFFFFFF. For values longer than eight hex characters, at least one separator blank is required after the eighth character.

If you do not set the PORTNAME parameter, it defaults to 0.

### **LUN** *hhhhhhhh hhhhhhhh*

The LUN value (*hhhhhhhh hhhhhhhh*) specifies the 8-byte logical unit number of the SCSI device that is IPLed. The value must be a hexadecimal value in the range 0 - FFFFFFFF FFFFFFFF. For values longer than eight hex characters, at least one separator blank is required after the eighth character.

If you do not set the LUN parameter, it defaults to 0.

### **BOOTprog** *bootprog\_number*

The *bootprog\_number* value specifies the boot program that is loaded from the SCSI or ECKD device that is IPLed. The value must be a decimal value in the range 0 - 30.

If you do not set the BOOTPROG parameter, it defaults to 0.

### **BOOTprog** **AUTOMATIC**

The BOOTPROG AUTOMATIC operand loads the first operating system program (not a dump program) that is listed in the program table of the device that is IPLed.

If you do not set the BOOTPROG parameter, it defaults to 0.

### **BR\_LBA** *hhhhhhhh hhhhhhhh*

The BR\_LBA value (*hhhhhhhh hhhhhhhh*) specifies the logical-block address of the boot record of the SCSI device that is IPLed. The value must be a hexadecimal value in the range 0 - FFFFFFFF FFFFFFFF. For values longer than eight hex characters, at least one separator blank is required after the eighth character.

If you do not set the BR\_LBA parameter, it defaults to 0.

### **BOOTREC** *cylinder head record*

The BOOTREC values (*cylinder head record*) specify the boot record location on the ECKD device that is IPLed. Use hexadecimal values and separate the values by blank space. Leading zeros are not required. The following ranges are allowed when you specify the boot record location:

- Cylinder: 0 - FFFFFFFF
- Head (or track): 0 - F
- Record: 1 - FF

### **BOOTREC** **LABEL**

The BOOTREC LABEL operands specify that the boot record is the record that is identified in the volume label of the ECKD device that is IPLed.

If you do not set the BOOTREC parameter, it defaults to LABEL.

### **SECURE**

#### **NOSECURE**

The SECURE operand specifies that the IPL is a secure IPL.

The NOSECURE operand specifies that the IPL is not a secure IPL.

If you do not set the SECURE or NOSECURE parameter, the parameter defaults to NOSECURE.

The SECURE parameter is not valid unless the DEVICE parameter (virtual device number) is set.

If a user's OPTION directory statement includes the SECUREIPLREQUIRED option, the SECURE operand must be set.

#### SCPDATA ... *text*

#### SCPDATA ... HEX *hex\_value*

The SCPDATA keyword designates data to be passed to the program that is loaded for a list-directed IPL. If the program does not require any data, then the SCPDATA parameter is optional.

The HEX keyword indicates that the data (*hex\_value*) is UTF-8 encoded hex (characters 0-F). The hex string must be an even number of digits.

If the HEX keyword is not specified, the data is assumed to be EBCDIC text (code page 924).

Quotation marks are necessary only if the text includes blanks, mixed case, is a valid keyword (NEW or APPEND), or is all numeric (a valid offset).

Up to 4096 (4K) characters of data (text or hexadecimal) can be entered. The actual number of input characters might be less than 4096 depending on the translation to UTF-8. Because two hex characters are required to represent each UTF-8 data byte, the maximum number of UTF-8 data bytes that can be defined by using the HEX option is 2048 (1/2 of the 4K character input limit).

There is no default value for SCPDATA.

#### NEW

The parameter value that follows the NEW keyword replaces any previously specified SCPDATA data.

#### APPEND

The parameter value that follows the APPEND keyword is appended to the end of any previously specified SCPDATA data.

#### offset

The *offset* value indicates an offset of the previously specified SCPDATA data. The parameter value that follows the *offset* value overlays the previously defined SCPDATA data beginning at the specified offset. The *offset* value must be a value from 0 to the current length of the SCPDATA data.

## Usage Notes

- LOADDEV and DUMPDEV parameters are used for list-directed IPLs. Table 36 on page 1547 shows which LOADDEV parameters have default values that must be set (changed from the default) before each type of list-directed IPL without dump can be started. Other parameters might need to be set if the default values are not appropriate for your environment.

IPL type	DEVICE parameter	PORTNAME parameter	LUN parameter	SECURE parameter
IPL <i>fcp_vdev</i>	The DEVICE parameter is not set or matches <i>fcp_vdev</i> on the <b>IPL</b> command	Required	Required	The SECURE parameter is not allowed when the <i>fcp_vdev</i> operand is used
IPL LOADDEV from SCSI device	Required	Required	Required	Required if the SECUREIPLREQ option is specified in the OPTION directory statement
IPL LOADDEV from ECKD device	Required	Not applicable	Not applicable	

## SET LOADDEV

Note that a list-directed IPL without dump uses LOADDEV parameters. A list-directed IPL with dump uses DUMPDEV parameters. For more information, see [“IPL” on page 566](#) and [“SET DUMPDEV” on page 1496](#).

2. To prepare for a secure IPL, the program that is loaded must be signed. The appropriate certificate must be loaded in the HMC and assigned to any LPAR on which the IPL might occur.
3. LOADDEV parameters can be set by either of the following methods:

- **SET LOADDEV** command.
- LOADDEV directory statement. The LOADDEV parameters that are set by the LOADDEV directory statement can be used at logon and persist after the logon. Those parameters can be used by a list-directed IPL of a running guest.

The LOADDEV user directory statement can be modified by a directory management tool such as IBM Directory Maintenance (DirMaint). To set load parameters by using DirMaint, use the **DIRM LOADDEV** command. The LOADDEV user directory statement can be modified also by the Image\_IPL\_Characteristics\_Define\_DM API function.

For more information, see **DIRM LOADDEV** in *z/VM: Directory Maintenance Facility Commands Reference* and **LOADDEV Directory Statement** in *z/VM: CP Planning and Administration*.

4. The LOADDEV settings of a user can be displayed by using the **QUERY LOADDEV** command. See [“QUERY LOADDEV” on page 985](#).
5. The initial **SET LOADDEV** command or the first **SET LOADDEV** command after a **SET LOADDEV CLEAR** command establishes the device type (SCSI or ECKD) until another **SET LOADDEV CLEAR** command is issued. Until another CLEAR is issued, **SET LOADDEV** commands with operands for a different device type are not valid. Because the default device type is SCSI, when you set load parameters for an ECKD device you must specify the ECKD operand on every **SET LOADDEV** command that is issued. For example:

```
SET LOADDEV CLEAR ECKD DEVICE 1234
SET LOADDEV ECKD BOOTREC 0 0 5
```

6. After a LOADDEV parameter is set, it can be changed by issuing the **SET LOADDEV** command with a new value for the same parameter. It is not necessary to issue a **SET LOADDEV CLEAR** command before you change a parameter value.

The one exception is the device type parameter (SCSI or ECKD). You must issue a **SET LOADDEV CLEAR** command before you change the device type parameter.

7. z/VM does not support list-directed IPL of a z/VM system from ECKD or a secure IPL of a z/VM system from SCSI or ECKD. Therefore, you cannot set the following LOADDEV parameters to IPL a second-level z/VM system:

- ECKD
- SECURE

To IPL a second-level z/VM system non-securely from a SCSI disk, specify the following information:

- Specify the virtual device number to IPL (DEVICE *fcv\_vdev*).
- Specify the world wide port name (PORTNAME).
- Specify the logical unit number (LUN).
- For the BOOTPROG operand, specify 0 (or let it default to 0).
- If the IPL device is a CP-formatted device, specify C8 for the BR\_LBA operand. The SALIPL utility writes the SCSI boot record to block 200 (X'C8') of a CP-formatted device.

For a CMS-formatted device, SALIPL writes the SCSI boot record at block 200 (X'C8') into the RECOMP area. Set BR\_LBA to the hexadecimal value of decimal 200 plus the number of blocks in the CMS area of the minidisk. The value is 200 plus the "noblk" value that is used on the CMS command to prepare the disk:

```
FORMAT vdev fm noblk (RECOMP
```

See example “4” on page 1549.

- Specify NOSECURE.
  - For the SCPDATA operand, optionally specify a console device number where you want the Stand Alone Program Loader screen to appear. The console device number can also be specified on the LOADPARM operand of the IPL command. Note that if both the SCPDATA operand and the LOADPARM operand are specified, the Stand Alone Program Loader uses the value that is specified by the LOADPARM operand.
8. The default LOADDEV value for the PORTNAME and LUN parameters is 0, which is not a valid device port name and not a valid device LUN. An IPL attempt that uses the default value for the PORTNAME or LUN parameter will fail.

## Examples

1. The following command identifies port a20, logical unit number 03 as the SCSI device, and passes the SCPDATA=x'74657374' to the program that is loaded during a guest IPL.

```
set loaddev port a20 lun 03 scpdata hex 74657374
```

2. After the command in the previous example completes, the following command replaces '57' in the SCPDATA with '75'. The offset of 3 indicates where the data overlay begins. The resulting SCPDATA value is 74675374.

```
set LOADDEV scpdata 3 hex 75
```

3. The following commands set the load device information for a SCSI disk that is CP-formatted and was initialized by using the SALIPL utility. The BR\_LBA value is the hexadecimal equivalent of 200.

```
set loaddev clear port 50050763 00ce04da
set loaddev lun 51460000 00000000 br_lba C8
```

4. The following commands set the load device information for a SCSI disk that is CMS-formatted and was initialized by using the SALIPL utility. It is a 50000 block minidisk with a RECOMP area that starts at block 48000. The BR\_LBA value is the hexadecimal equivalent of 48000 + 200. The Stand Alone Program Loader screen appears on the console at device number 9.

```
set loaddev clear port 50050763 00ce04da
set loaddev lun 51460000 00000000 br_lba bc48 scpdata new 09
```

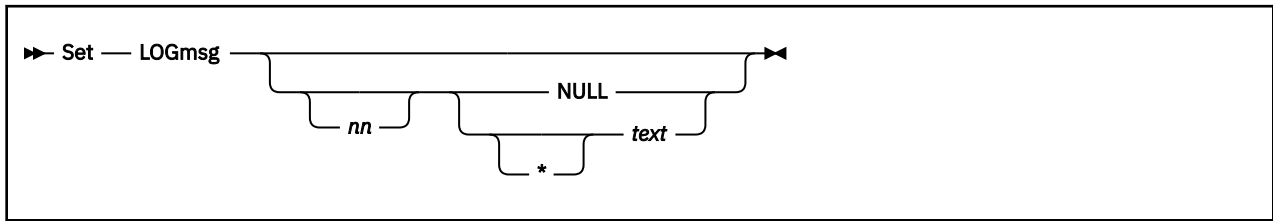
## Messages

- HCP002E Invalid operand - *operand*
- HCP021E A device number was not supplied or it is invalid.
- HCP1613E An attempt was made to set both SCSI and ECKD parameters for a future IPL. These are conflicting parameters.
- HCP2768E Missing cylinder number
- HCP2768E Missing track number
- HCP2768E Missing record number
- HCP2813E CP is running on a processor that does not support guest secure IPL.
- HCP2813E One or more systems in your relocation domain do not support guest secure IPL.
- HCP2813E CP is running on a processor that does not support list-directed IPL from ECKD devices.
- HCP2813E One or more systems in your relocation domain do not support list-directed IPL from ECKD devices.
- HCP2813E CP is running on a processor that does not support BOOTPROG AUTOMATIC.
- HCP2813E One or more systems in your relocation domain do not support BOOTPROG AUTOMATIC.

## SET LOADDEV

- HCP2825E SCPDATA format conflicts with previous input
- HCP2826E SCPDATA contains invalid UTF-8 data: *data*
- HCP2827E SCPDATA offset exceeds current length of data (*length*)
- HCP2828E SCPDATA length exceeds maximum allowed (*maximum*)
- HCP2829E SCPDATA does not contain a valid hex string
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*
- HCP6706E Invalid number - *number*
- HCP6706E Invalid cylinder number - *cylinder*
- HCP6706E Invalid track number - *track*
- HCP6706E Invalid record number - *record*

## SET LOGMSG



### Authorization

Privilege Class: B

### Purpose

Use SET LOGMSG to:

- Set the log message line you specify to the text you enter on the command line.
- Delete a specified line from the log message.
- Delete the entire log message.

The SET LOGMSG setting is temporary and disappears with any reset other than a warm start.

**Note:** If you enabled the file-based LOGMSG support using the FEATURES statement in the system configuration file, you cannot use the SET LOGMSG command to alter the log message. In this case, you need to edit log message files on CP-accessed minidisks and notify CP of the changes using the CPACCESS command.

### Operands

#### *nn*

is the line number in the log message you want changed. The variable *nn* is a decimal number between 0 and 50.

#### \* *text*

is the text that you want placed at line *nn* of the log message. If you want the line automatically displayed on each user's console during logon, you must precede the text with an asterisk (\*). The maximum number of bytes per line is 76, which includes blanks and the asterisk, if specified. In DBCS, if *text* is longer than 76 bytes, CP truncates the text to 76 bytes, but the SO/SI pair will be maintained. You can use mixed DBCS strings as message text. If line *nn* already has text, it is replaced. Otherwise, the text is inserted for line *nn*.

#### NULL

indicates that the current text for line *nn* is deleted from the log message. If NULL is not preceded by a line number, the entire log message is deleted.

### Usage Notes

1. CP only sends the response and redisplayed input lines to the terminal, not to the virtual machine.
2. If you do not specify any operands after SET LOGMSG, the system prompts you to enter a line of text. This line of text is inserted into the log message after the last line that already has text. For example, if the highest line number in the log message to have text is 37, the new line of text is inserted at line 38. If line 50 already has text defined, you receive an error message and the text is not inserted.
3. The maximum number of lines you can include in your log message is 50, including blank lines; the highest line number in the log message is 50.

## SET LOGMSG

4. If you specify a line number with nothing following it (for example, SET LOGMSG *nn*), the system prompts you to enter a line of text. The text entered is placed in the log message at the line specified in the command. If you press ENTER without entering any text, the line is deleted from the log message.
5. Enter SET LOGMSG 0 to allow yourself to insert a new line into the log up to a maximum of 50 lines.
6. If the text of a line in the log message does not start with an asterisk (\*), the user must enter QUERY LOGMSG to see the text. (The text is not displayed automatically when the user logs on to the system.)
7. If more than 76 characters of text are entered for a line, the text is truncated after the 76th character without any message being issued.
8. The LOGMSG area is checkpointed at system termination.

### Responses

Response 1:

```
LOGMSG:
```

prompts you to enter a line of text when you enter either SET LOGMSG or SET LOGMSG *nn*. Type the desired line of text in response to the prompt and press the ENTER key.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP041E Logmsg *nn* does not exist.
- HCP1506E The last line (50) in the log message is already in use.



## SET LSYSTEM

```
▶ SET — LSYStem — lsysname ▶
```

### Authorization

Privilege Class: B

### Purpose

Use SET LSYSTEM to set or change the machine name of the local system. This name is used when identifying two machines connected by CIB (coupling over InfiniBand) channel paths.

### Operands

#### LSYStem *lsysname*

is the machine name of the local system. The variable *lsysname* must be a 1- to 8-character alphanumeric string.

### Usage Notes

1. CP must be enabled to allow dynamic-I/O configuration changes (by the FEATURES ENABLE DYNAMIC\_I/O statement in the system configuration file or the SET DYNAMIC\_I/O operator command) in order for this command to be allowed.
2. When you execute this command, the machine enters and exits configuration mode.
3. This command cannot be issued while the machine is in configuration mode.
4. If you are going to use the SET CONFIG ON command to make dynamic-I/O changes that involve the use of the local system name, and you also want to change that name, use this SET LSYSTEM command to change the name before using the SET CONFIG ON command.
5. The local system name can be defined in your IOCP configuration definition using the LSYSTEM keyword on the ID statement. You must define this name in the IOCP definition if CIB-type channel paths are defined.

### Responses

Response 1:

To set your local system name to CPCSYS43, enter the following:

```
set lsys cpcsys43
The local system name was set as requested.
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP1006E Command processing cannot complete because the system does not support the coupling over InfiniBand channel-path facility.
- HCP6545E HCD currently controls the hardware configuration. All CP hardware configuration commands are disabled.
- HCP6704E Missing token at end of line.

## SET LSYSTEM

- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request failed with return code *nnnn*
- HCP6808E Cannot find the configuration token on the system
- HCP6808E System disabled for dynamic I/O changes
- HCP6808E The configuration mode setting of the hardware and software do not match
- HCP6808E The dynamic I/O support was disabled at the service console
- HCP6808E The VM configuration token does not match the hardware configuration token
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in channel subsystem.
- HCP6815E Cannot find configuration token in the system
- HCP6816E Dynamic I/O changes are not allowed on this system
- HCP6819E A dynamic I/O command is currently executing for user *userid*
- HCP6850E Cannot set local system name. A dynamic I/O change is in progress.
- HCP6850E Cannot set local system name. System is in configuration mode.
- HCP6851E I/O configuration cannot be changed because VM is not the controlling partition

## SET MACHINE



### Authorization

Privilege Class: G

### Purpose

Use SET MACHINE to change the architecture in effect for your virtual machine. When you select an architecture mode, your virtual machine must obey that particular mode's architecture conventions.

### Operands

#### ESA

designates an ESA virtual machine, which can simulate either ESA/390 architecture or z/Architecture. See [Table 37 on page 1555](#). Also see usage note “7” on [page 1556](#).

#### XA

designates an XA virtual machine, which is functionally equivalent to an ESA virtual machine.

#### XC

designates an XC virtual machine, which simulates ESA/XC architecture and which can be switched into z/XC architecture.

#### Z

designates a Z virtual machine, which simulates a z/Architecture-only environment.

*Table 37. Virtual Machine Architecture Modes*

Virtual Machine Designation	Virtual Machine Architecture	Address Width	Addressable Storage
ESA, XA	ESA/390 or z/Architecture <sup>1</sup>	31-bit or 64-bit <sup>1</sup>	2 GB or 16 EB <sup>1</sup>
XC	ESA/XC or z/XC <sup>2</sup>	31-bit or 64-bit <sup>2</sup>	2 GB or 16 EB <sup>3</sup>
Z	z/Architecture	64-bit	16 EB

#### Note:

1. An ESA virtual machine simulates ESA/390 architecture when initially defined. However, a guest can switch the virtual machine between ESA/390 and z/Architecture mode. A z/Architecture virtual machine supports 64-bit addressing and addressable storage beyond 2 GB.
2. An XC virtual machine simulates ESA/XC architecture when the virtual machine is initially defined. However, a guest can switch the virtual machine between ESA/XC and z/XC architectures. z/XC supports 64-bit addressing and storage beyond 2GB.
3. Data spaces are limited to 2 GB.

## Usage Notes

1. If the SET MACHINE command changes your virtual machine architecture mode, a system-reset-clear function is performed on your virtual machine as if you had issued a SYSTEM CLEAR command. The effects of a system-reset-clear function are explained in detail in the description for the SYSTEM CLEAR command. Because a system-reset-clear function sets all of your virtual machine's storage to binary zeros, you must repeat your IPL procedure to continue operation.  
  
If you enter the command but it does not change your virtual machine architecture mode, no system-reset-clear function is performed on your virtual machine, and your virtual machine's environment remains as it is.
2. If the SET MACHINE command places your virtual machine in the ESA/XC architecture mode, the time-of-day (TOD) clock for your virtual machine is reset to match the TOD clock on the real processor.
3. A default virtual machine architecture mode can be specified on the user directory MACHINE statement. If no machine statement is used, the machine definition on the GLOBALOPTS statement of the directory is used. If no GLOBALOPTS statement is used, the default is XA.
4. For compatibility, z/VM continues to accept the virtual machine designation XA and continues to report XA as the virtual machine type in response to QUERY SET and INDICATE USER commands when the virtual machine has been defined using the XA designation. However, whether the virtual machine is defined using SET MACHINE XA or SET MACHINE ESA makes no difference when running on z/VM. A virtual machine defined with SET MACHINE XA has the capabilities of an ESA virtual machine and is considered by CP to be an ESA virtual machine.
5. CP does not support System/370 mode virtual machines. A SET MACHINE 370 command will be rejected with message HCP1016E.
6. If an attempt is made to SET MACHINE Z by a guest whose relocation domain does not support the facility, the command will be rejected with message HCP1016E.
7. An ESA (or XA) virtual machine will be put into either full ESA/390 mode or ESA/390 compatibility mode, depending on the level of ESA/390 capability available in the machine where the virtual machine is logged on or in the virtual machine's relocation domain. ESA/390 compatibility mode allows a subset of ESA/390 functionality sufficient for CMS and GCS, but lacks advanced functions like dynamic address translation (DAT). For details on ESA/390 compatibility mode, see [z/Architecture Principles of Operation](#).

## Responses

Response 1:

```
SYSTEM RESET
```

indicates that the virtual machine has been cleared.

Response 2:

```
SYSTEM =[ESA]
         [XA ]
         [XC ]
         [Z  ]
```

### ESA

indicates that the virtual machine mode has been set to ESA.

### XA

indicates that the virtual machine mode has been set to XA.

### XC

indicates that the virtual machine mode has been set to XC.

### Z

indicates that the virtual machine mode has been set to Z.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP1016E Command rejected; CP does not support System/370 mode virtual machines.
- HCP1016E Command rejected; SET MACHINE Z is not available in the current relocation domain.

## SET MAXLDEV

---

```
▶▶ Set — MAXLdev — nnnnn ▶▶
```

### Authorization

Privilege Class: E

### Purpose

Use SET MAXLdev to change the maximum number of logical devices that can exist at the same time.

### Operands

#### *nnnnn*

is the maximum number of logical devices allowed at one time. The default is 4096. The range is from 4096 to 32768.

### Usage Notes

1. The count of logical devices on the system may be greater than the maximum specified. This can happen when the limit is lowered to a value less than the current number of logical devices in existence. If this occurs, the existing logical devices will be unaffected, but no new logical devices can be created until the number of logical devices is less than the maximum allowed.

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid

## SET MAXUSERS



### Authorization

Privilege Class: A

### Purpose

Use SET MAXUSERS to control the number of users able to log on.

### Operands

#### *nnnnn*

is the maximum number of users allowed to log on. The range is between 1 and 99,999.

#### **OFF**

indicates that a maximum number of logged on users has not been explicitly set, so the system default limit of 99,999 is in effect.

### Usage Notes

1. If the maximum number of users is set to a number lower than the current number of logged-on users, the system allows the users to log off, to be forced off the system, and to disconnect and reconnect. However, it does not allow any users to log on until the current number of logged on users is lower than the number specified by MAXUSERS unless the user directory for the virtual machine that is logging on was created with the IGNMAXU operand of the OPTION control statement.

### Responses

Response 1:

```
MAXUSERS = nnnnn
```

is the normal response for this command.

Response 2:

```
REQUEST ACCEPTED. MAXIMUM USERS VALUE IS CURRENTLY EXCEEDED
```

is returned when the number of users already logged on exceeds the limit just set.

Response 3:

```
THE MAXUSERS LIMIT CHECKING HAS BEEN TURNED OFF
```

is returned when SET MAXUSERS OFF is entered and the maxusers limit was on. The response means the system default of 99,999 users is now in effect.

Response 4:

```
NO MAXUSERS LIMIT IS SET
```

is returned when SET MAXUSERS OFF is entered and the maxusers limit was already off. The response means the system default of 99,999 users is now in effect.

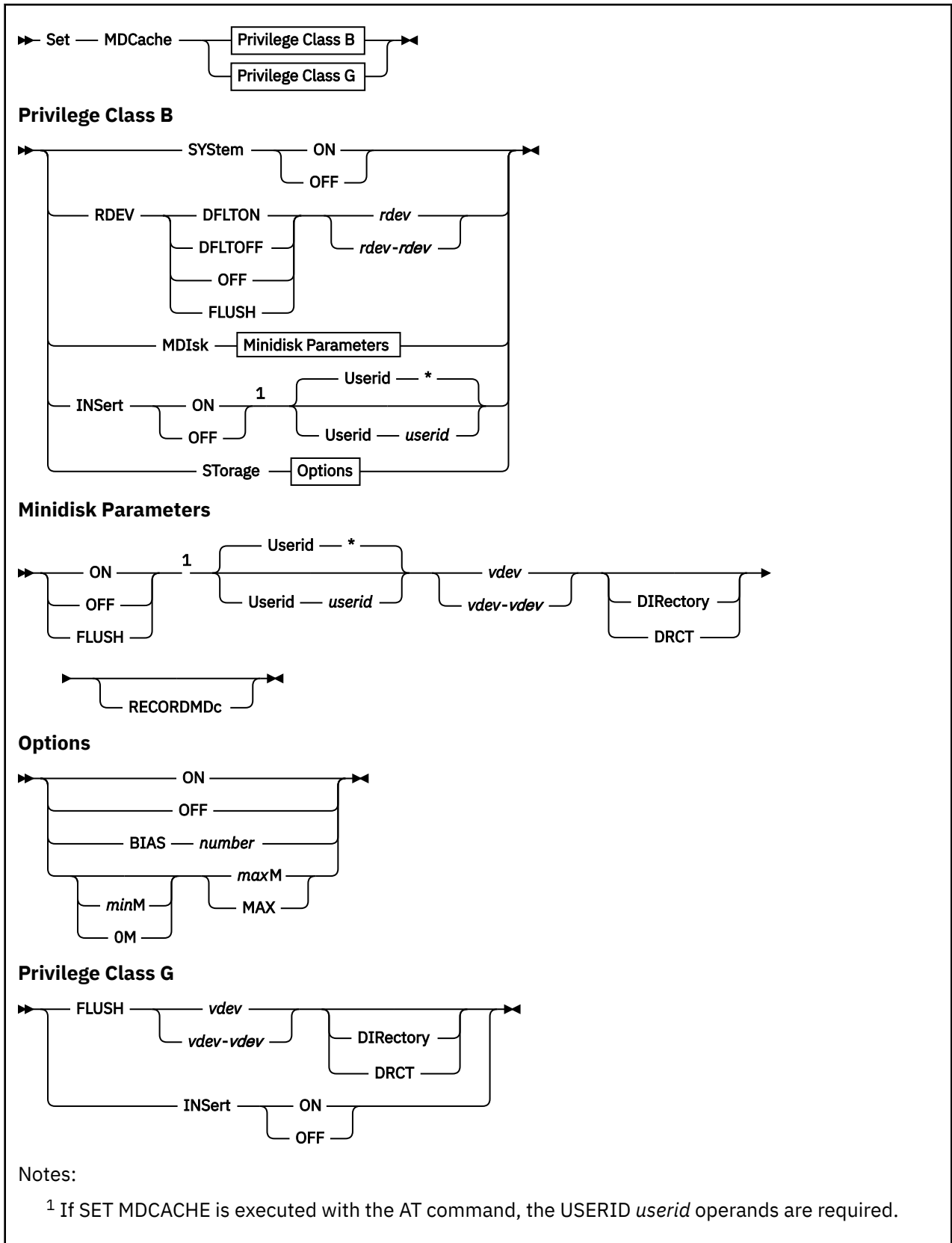
## SET MAXUSERS

### Messages

- HCP002E Invalid operand - *operand*
- HCP1001E An operand is missing for *option*.



# SET MDCACHE



## Authorization

Privilege Class: B, G

## Purpose

Use SET MDCache to change minidisk cache settings or change a user's ability to insert data into the cache.

A class B user can:

- Set caching on or off for all minidisks accessed on this system.
- Control minidisk caching at the real device or minidisk level.
- Purge the cache of data from a real device or an active minidisk.
- Change any user's ability to insert data into the cache.
- Specify the maximum and minimum amounts of real storage that can be used by a minidisk cache. The actual size of the minidisk cache varies and is determined dynamically by the paging and I/O load.

A class G user can:

- Purge the data in the cache pertaining to an active minidisk or a minidisk defined in the user's directory entry.
- Change the user's ability to insert data into the cache.

## Operands

### SYSTEM

### RDEV

### MDisk

defines the hierarchy level to which the command pertains, which determines whether a given virtual I/O will be cached.

### SYSTEM

specifies the highest level of control of the minidisk cache. With the SYSTEM option, minidisk caching can be globally turned on (enabled) or off (disabled) for the entire system via SET MDCACHE SYSTEM ON and SET MDCACHE SYSTEM OFF. If minidisk caching is on at the system level, it can be further controlled at the real device and minidisk level.

### RDEV

specifies the next highest level of control of the cache. If minidisk caching is on for the system, the installation can change the cache eligibility for a given real device, and hence, all minidisks that reside on that device via SET MDCACHE RDEV *action rdev* (or SET MDCACHE RDEV *action rdev-rdev*) where *action* is either DFLTON, DFLTOFF, OFF, or FLUSH.

### MDisk

specifies the lowest level of control of the cache. If minidisk caching is on for the system, the installation can change the cache eligibility for an active minidisk via SET MDCACHE MDISK *action USERID userid vdev* (or SET MDCACHE MDISK *action USERID userid vdev-vdev*). Note that for a minidisk to be cached, the system must be enabled for caching, the real device on which it resides must be set to DFLTON or DFLTOFF for caching, and the minidisk itself must be enabled for caching.

Class G users can use the SET MDCACHE FLUSH *vdev* (or SET MDCACHE FLUSH *vdev-vdev*) to purge the cache of data pertaining to minidisks defined in their directory.

The duration of cache control at the minidisk level is not permanent when set by the SET MDCACHE command. Permanent cache control at the minidisk level is achieved by the use of the MDC and NOMDC options of the MINIOPT directory statement. For the SET MDCACHE command to be successful for a minidisk, the minidisk must be active (that is, the minidisk must be linked by at least one user who is logged on the system). If at a later time, all links to the minidisk are

dropped, then the cache eligibility of the minidisk is determined from the directory entry for any subsequent link to the minidisk.

**ON**

is valid at the SYSTEM and MDISK levels.

- At the SYSTEM level, the ON option enables minidisk caching for the entire system without changing any real device or minidisk cache settings.
- At the MDISK level, the ON option enables minidisk caching for a minidisk. Note that the real device on which the minidisk resides must be set to DFLTON or DFLTOFF also before caching will occur. If specified without the RECORDMDC option and record caching is enabled, record caching is turned off for the minidisk.

**OFF**

is valid at the SYSTEM, RDEV, and MDISK levels.

- At the SYSTEM level, the OFF option disables minidisk caching for the entire system without changing any real device or minidisk cache settings.
- At the RDEV level, the OFF option disables minidisk caching for an entire real device. This command overrides any minidisk MDC setting set by the MINIOPT directory statement or the SET MDCACHE MDISK command.
- At the MDISK level, the OFF option disables minidisk caching for a minidisk. It also turns off RECORDMDC for the minidisk if it is active.

**DFLTON**

is valid at the RDEV level.

The DFLTON option enables minidisk caching for an entire real device, yet allows the installation to disable cache eligibility for any minidisk that resides on that device via the MINIOPT NOMDC directory statement or the SET MDCACHE MDISK OFF command

**DFLTOFF**

is valid at the RDEV level.

The DFLTOFF option disables cache eligibility for the entire real device, yet allows the installation to enable cache eligibility for any minidisk that resides on that device via the MINIOPT MDC directory statement or the SET MDCACHE MDISK ON command.

**FLUSH**

is valid at the RDEV and MDISK levels.

- At the RDEV level, the FLUSH option temporarily purges the data in the cache belonging to a real device. It does not change cache eligibility settings.
- At the MDISK level, the FLUSH option temporarily purges the data in the cache belonging to a minidisk. It does not change cache eligibility settings. Class G users can use the SET MDCACHE FLUSH *vdev-vdev* command for active minidisks defined in their directory entry.

***rdev******rdev-rdev***

identifies the real device numbers whose cache eligibility is to be changed.

**Userid \*****Userid *userid***

is the user ID of the virtual machine for which you want to change minidisk cache settings. If omitted or specified as an asterisk (\*), CP uses the user ID of the virtual machine issuing the command. For the class B command, if DRCT or DIRectory is not specified, the userid specified must be logged on.

***vdev******vdev-vdev***

identifies the virtual device number or the range of virtual device numbers for which cache settings are to be changed. For the class B command, the virtual device numbers specified correspond to the userid's current configuration if the DIRECTORY or DRCT option is not specified and to the userid's directory entry if the DIRECTORY or DRCT option is specified.

**DIRectory****DRCT**

indicates that the virtual device numbers correspond to the userid's directory entry. For the class B command, if DRCT or DIRectory is not specified, the userid specified must be logged on.

**RECORDMDC**

indicates that the minidisk's data should be cached using 4 KB record caching instead of normal full track caching. Only non-FBA 4 KB CMS formatted I/O via Diagnose X'18', X'A4', X'250', and \*BLOCKIO CP System Service is cached. All other I/O results in I/O direct to the minidisk. This option is only valid for SET MDCACHE MDISK ON.



**Attention:** Use of this option is a last resort for very special and unusual circumstances. It should only be used as the result of consultation with IBM support personnel who have concluded that no other performance problem exists. Using this option for any other reason may mask a performance problem which when corrected allows normal full track minidisk cache to perform better than with the RECORDMDC option.

**INSert**

changes a user's ability to insert information into the cache.

- The ON option allows a user to insert data into the cache.
- The OFF option prevents a user from inserting data into the cache. The user will continue to use data that is already in the cache, but cannot insert data. This option is useful when doing a scan of multiple files to avoid corrupting the cache with data that will only be referenced once.

**STorage**

used to control the size of the real storage portion of the minidisk cache or enable/disable caching to real storage.

**ON**

indicates that CP uses the amount of minidisk cache that best balances I/O and paging performance. This is the default. The minidisk cache size can range from 0 to the size of real storage, although the actual maximum is really limited to the size of the dynamic paging area (DPA).

**OFF**

indicates that CP does not use any of the identified storage for minidisk caching. The minidisk cache size is set to 0 for the specified storage type.

**BIAS number**

designates a number (value) that the MDC arbiter will use to adjust the target size for the indicated storage. The number must be within the range .01 to 100.00. For whole numbers, the decimal point is optional. The arbiter will multiply its selected target size by number and use the product (subject to the specialized min and max ranges) as the new target size. Therefore, if the number is greater than one, the effective target size will be larger than the one chosen by the arbiter and if the number is smaller, the target will be correspondingly smaller.

**OM****minM**

indicates the minimum size of the minidisk cache in megabytes if OM or minM. If *min* is greater than the amount of real storage on the system, the value is decreased to the amount of real storage on the system. If *minM* is specified, *maxM* must also be specified. If *minM* is not specified, the current minimum does not change unless it is now larger than the maximum in which case it is set to equal the maximum.

**MAX****maxM**

indicates the maximum storage size in megabytes for the indicated storage. If *max* is greater than the amount of real storage on the system, the value is decreased to the amount of real storage on the system. If *minM* is specified, *max* must be greater than or equal to *min*. If MAX is specified for the maximum real storage size, the upper limit of the real storage portion of the minidisk cache is set to the size of online real storage.

If the value specified as *max* is 0, the minidisk caching is turned off for that type of storage. Setting the maximum value to 0 is likely to degrade performance. The smallest recommended value is 2.

When *min* equals *max*, the size of the minidisk cache is fixed for that type of storage. It does not vary dynamically.

## Usage Notes

1. In z/Architecture mode, minidisk cache uses real storage above the 2 GB line.
2. If the amount specified via SET MDCACHE STORAGE is greater than the amount available, the amount specified is rounded down to the amount available.
3. Issuing SET MDCACHE STORAGE OFF or SET MDCACHE SYSTEM OFF prevents caching from occurring on the system. However, issuing SET MDCACHE SYSTEM OFF immediately purges all cache data, disables caching on the system, cleans up all internal minidisk cache structures, and releases all storage used by caching. Issuing SET MDCACHE STORAGE OFF causes a gradual reduction in cache data and does not clean up internal structures or storage. Furthermore, caching is still considered enabled for the system.
4. When SET MDCACHE STORAGE is issued with any option other than OFF, minidisk caching is enabled.
5. If SET MDCACHE STORAGE *maxM* is issued when *max* is greater than the current size of the minidisk cache in real storage, a gradual reduction in the use of real storage for CP paging occurs until the minidisk cache attains the new size. This does not cause a rapid migration of paging data to auxiliary storage. Similarly, when *max* is less than the current size of the minidisk cache, a gradual reduction in the use of minidisk cache occurs until the minidisk cache attains the new size.
6. Using the DIRECTORY or DRCT options is highly recommended as the user's current virtual configuration could vary between the time a user's virtual device is queried and the time the SET MDCACHE command is issued. The SET MDCACHE command could end up being issued for the wrong virtual device.
7. It is recommended that if minidisk caching is used, STORAGE should not be set to OFF. Even if it is set to OFF, up to 2 MB of real storage can be used for minidisk caching.
8. A minidisk, including a temporary disk, is not eligible for minidisk cache if it has been defined with greater than 32767 cylinders. This does not apply to FBA devices. It applies only to ECKD devices.
9. The RECORDMDC option of the SET MDCACHE MDISK ON command should only be used when directed to do so by IBM support personnel. RECORDMDC is for use only when DASD cache as well as real storage minidisk cache are all being overwhelmed by excessive unreferenced records being read by the full track cache support.
10. Use the SET MDCACHE INSERT command to temporarily disable minidisk cache operation for a minidisk (for example: during backup for the minidisk). Do not use SET MDCACHE MDISK as this causes unnecessary overhead and may cause a change in the caching mode for this minidisk.
11. The SET MDCACHE command is not valid for an alias Parallel Access Volume. Cache settings for an alias Parallel Access Volume are inherited from its associated base Parallel Access Volume.
12. The SET MDCACHE command is not valid for an alias HyperParallel Access Volume. Cache settings are only applicable for base HyperParallel Access Volumes.
13. When SET MDCACHE RDEV ON or SET MDISK ON is issued, a minidisk will not be eligible for cache until all write links on other systems in the SSI cluster are dropped, if any exist.

## Responses

Response 1:

The following response is issued when you specify SET MDCache SYStem ON.

```
Minidisk cache ON for system
Storage MDC min=nnnM max=nnnM, usage=ppp%, bias=b.bb
```

**min=nnnM**

indicates the current minimum value for the size of minidisk cache in megabytes.

## SET MDCACHE

### **max=nnnM**

indicates the current maximum value for the size of minidisk cache in megabytes.

### **usage=ppp%**

indicates the percentage of real storage in use by minidisk cache.

### **bias=b.bb**

indicates the bias factor used by the arbiter to determine the target size for the storage type.

Response 2:

The following response is issued when you specify SET MDCACHE SYSTEM OFF.

```
Minidisk cache OFF for system
```

Response 3:

The following response is issued when you specify SET MDCACHE RDEV DFLTON/DFLTOFF/OFF/FLUSH *rdev*.

```
Minidisk cache {DFLTON } for rdev
                {DFLTOFF}
                {OFF     }
                {FLUSHED}
```

### ***rdev***

identifies the real device number.

Response 4:

The following response is issued when you specify SET MDCACHE RDEV DFLTON/DFLTOFF/OFF/FLUSH for a range of real device numbers.

```
Minidisk cache {DFLTON } for rdev-rdev
                {DFLTOFF}
                {OFF     }
                {FLUSHED}
```

### ***rdev-rdev***

identifies the first and last real device numbers in the range.

Response 5:

The following response is issued when you specify SET MDCACHE MDISK ON/OFF/FLUSH USERID *userid* *vdev* or the class G command SET MDCACHE FLUSH *vdev*.

```
Minidisk cache {ON      } for userid vdev [recordmdc enabled]
                {OFF     }
                {FLUSHED}
```

### ***userid***

identifies the userid.

### ***vdev***

identifies the virtual device number.

### **recordmdc enabled**

indicates that record level minidisk caching is in effect for this minidisk rather than normal full track minidisk caching.

Response 6:

The following response is issued when you specify SET MDCACHE MDISK ON/OFF/FLUSH FOR USERID *userid vdev-vdev* or the class G command SET MDCACHE FLUSH *vdev*.

```
Minidisk cache {ON      } for userid vdev-vdev [recordmdc enabled]
                {OFF     }
                {FLUSHED}
```

**userid**

identifies the userid.

**vdev-vdev**

identifies the first and last virtual device numbers in the range.

**recordmdc enabled**

indicates that record level minidisk caching is in effect for this minidisk rather than normal full track minidisk caching.

Response 7:

The following response is issued when you specify SET MDCACHE INSERT ON/OFF.

```
Minidisk cache inserts {ON } for userid
                       {OFF }
```

**userid**

identifies the userid whose insert capability has been changed.

**Messages**

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP0013E Conflicting option - *option*
- HCP0026E Operand missing or invalid
- HCP0045E *userid* not logged on
- HCP1401I Expanded storage is not supported.
- HCP1920E Minidisk cache not enabled for system
- HCP1921E Cache settings could not be {changed | queried} for {*rdev code cc* | *userid vdev code cc* | *rdev-rdev* | *userid vdev-vdev* | system code *cc*}
- HCP6630E Minidisk Cache will be enabled later for *userid vdev*; write links exist in the SSI cluster.

**Note:** In message HCP1921E, the meanings of the value *cc* are:

**4**

device is not attached to system or not a cacheable device type

**8**

device does not exist

**12**

information could not be obtained from the directory

**16**

userid not in the directory

**20**

not enough storage to initialize cache

**24**

start or end of FBA minidisk not on full page boundary

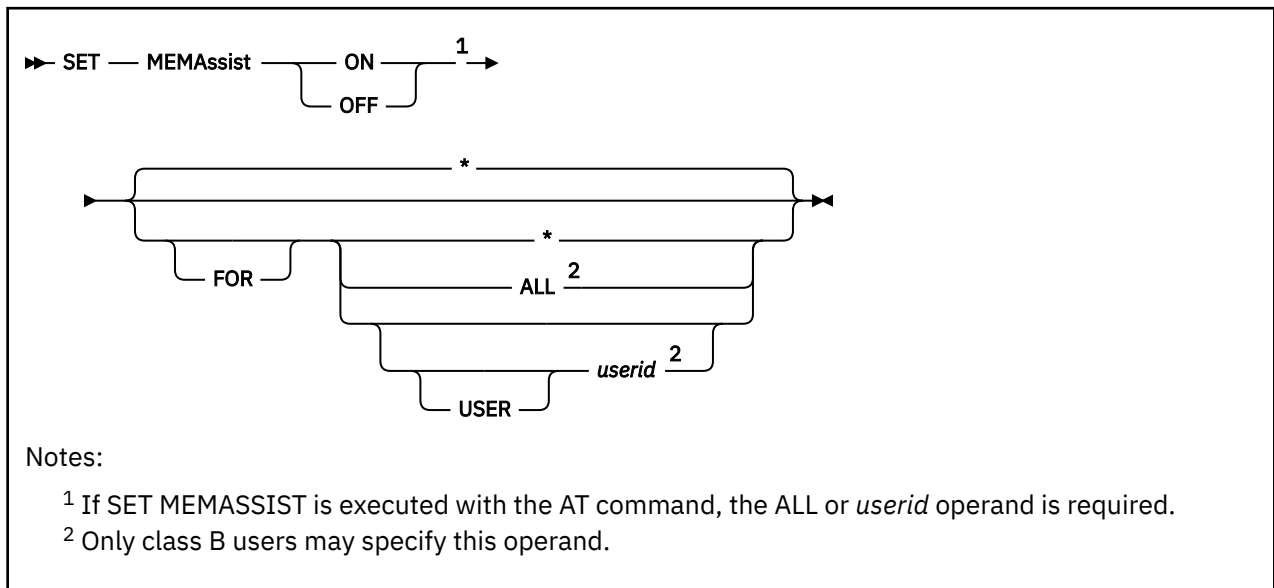
**28**

device is an alias Parallel Access Volume. Cache settings are inherited from its associated base Parallel Access Volume.

**32**

device is an alias HyperParallel Access Volume. Cache settings are only applicable for base HyperParallel Access Volumes.

## SET MEMASSIST



### Authorization

Privilege Class: B, G

### Purpose

Use SET MEMASSIST to control the collaborative memory management assist for a virtual machine. This interpretive-execution assist applies to guests that use the EXTRACT AND SET STORAGE ATTRIBUTES (ESSA) instruction.

The following are required to use the collaborative memory management assist:

- The assist must have been permitted for the system, either by default or with the MEMASSIST ON FOR ALL setting.
- The assist must not be forbidden for the virtual machine with the MEMASSIST OFF setting.

### Operands

#### ON

#### OFF

enables or disables the assist for a virtual machine.

#### \*

indicates you want to set your own virtual machine's MEMASSIST setting.

#### ALL

permits or disallows eligibility for all virtual machines to use the assist. Current individual virtual machine settings are not changed. When ALL is specified by a class B user, all new accesses to the assist are permitted or disallowed for all virtual machines, including any new users that might log on. See usage notes “3” on page 1569 and “4” on page 1569.

#### USER *userid*

specifies the user ID of the specific virtual machine for which you want to set the assist.

### Usage Notes

1. The initial MEMASSIST settings for your virtual machine are determined as follows:



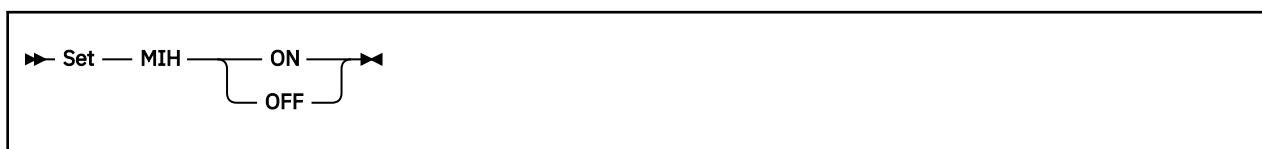
- a. If the collaborative memory management assist is installed on the system, the initial setting will be ON unless the NOMEMASSIST directory option is specified.
  - b. If the assist is not installed, the initial setting is OFF.
2. At z/VM IPL, the initial setting for MEMASSIST FOR ALL is ON, even if the assist is not installed on the machine. This allows a virtual machine to SET MEMASSIST ON, causing the assist to be simulated by z/VM for guest testing purposes.
  3. If your virtual machine's MEMASSIST setting is ON, then SET MEMASSIST OFF does not immediately disable the assist. Instead, the assist is placed into a PENDING OFF state and is disabled at the next virtual machine reset (such as a re-IPL).
  4. A SET MEMASSIST OFF FOR ALL command does not affect any user who might be currently using the assist. The command merely prevents any user from issuing the SET MEMASSIST ON command, and places users with MEMASSIST ON into a PENDING OFF state.
  5. You can use the class G QUERY MEMASSIST command to inspect the current MEMASSIST setting for your own virtual machine. You can use the class B QUERY MEMASSIST command to inspect another virtual machine's MEMASSIST settings. For more information, see [“QUERY MEMASSIST” on page 1008](#).
  6. If the collaborative memory management assist is not installed on the machine, guests can still use it by means of the simulation that is built into CP. However, because of the overhead incurred by simulation, this mode of operation is recommended for testing purposes only.

## Messages

- HCP003E Invalid option - *option*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP1014E A required option is missing
- HCP2163I MEMASSIST is not permitted

## SET MIH

---



### Authorization

Privilege Class: G

### Purpose

Use SET MIH to specify whether missing interrupt conditions should be handled for I/O operations involving devices dedicated to your virtual machine.

### Operands

#### ON

indicates that CP automatically handles missing interrupt conditions to the virtual machine's dedicated devices.

#### OFF

indicates that the missing interrupt conditions to the virtual machine's dedicated devices are detected but are not handled.

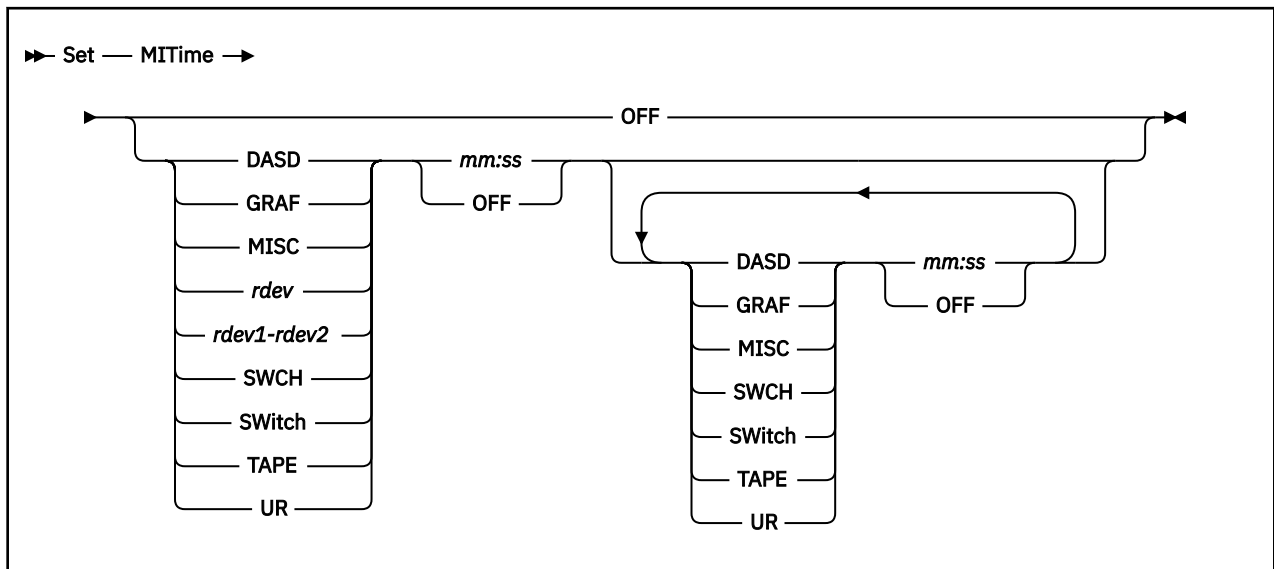
### Usage Notes

1. SET MITIME OFF overrides any setting of the MIH. Missing interrupts can be handled only if they are detected.
2. The default, which is to not handle missing interrupt conditions detected for the virtual machine's dedicated devices, is in effect when MIH is not specified on the user directory OPTION statement or the SET command is not issued.

### Messages

- HCP026E Operand missing or invalid

## SET MITIME



### Authorization

Privilege Class: A, B

### Purpose

Use SET MITIME to control the time interval at which the device specified is checked for missing interrupts.

### Operands

#### DASD

specifies that all real devices generated as DASD during system generation are to be monitored using the new time interval. Devices used by the mass storage subsystem (MSS) are not included.

#### GRAF

specifies that all real devices generated as display devices during system generation are to be monitored with the new time interval, an exception being device TYP328X.

#### MISC

specifies that real devices TYP328X, TYP3800, and any mass storage device are to be monitored with the new time interval.

#### *rdev*

#### *rdev1-rdev2*

is a real device number or range of real device numbers that are to be monitored with the new time interval.

#### SWCH

#### SWitch

specifies that all real devices generated as switch devices during system generation are to be monitored with the new time interval.

#### TAPE

specifies that all real devices generated as tape devices during system generation are to be monitored with the new time interval.

## SET MITIME

### UR

specifies that all real devices generated as spooling devices during system generation are to be monitored with the new time interval, an exception being device TYP3800.

### mm:ss

specifies the rate in minutes and seconds at which the device is to be examined for missing interrupts. The maximum value for *mm* is 99 and for *ss* is 59. The value is rounded up to the next multiple of five seconds.

### OFF

specifies that missing interrupts are not to be detected for any devices within the specified range. If you specify this operand without any device, missing interrupt monitoring for all devices is stopped.

## Usage Notes

1. If z/VM is running virtually, missing interrupt detection defaults to OFF at IPL. The default value is SET MITIME OFF.

When a real and a virtual system share a display device, the virtual system should leave missing interrupt detection off for the shared display device. Under certain circumstances, if the missing interrupt detector is active in the virtual machine for the shared display device, the virtual machine operator may be logged off.

2. The system default values at IPL time are:

<u>Device</u>	<u>Default Value</u>
SET MITIME DASD	00:15*
SET MITIME GRAF	01:10
SET MITIME UR	01:00
SET MITIME TAPE	10:00
SET MITIME SWITCH	05:00
SET MITIME MISC	12:00

\*The DASD and TAPE system default values will be used only if the device subsystems do not return a primary time-out value for the missing interrupt handler.

3. If you specify a range of devices, *rdev-rdev*, it must be in ascending sequence; otherwise, the command is invalid.
4. If you specify a value for *:ss* that is greater than 59, the command is considered not valid. If you specify 00:00, missing interrupt detection for the devices is stopped.
5. It may take up to 5 seconds longer than the interval specified to detect a missing interrupt.
6. When SET MITIME is issued, the new time interval is applied to any current operation.
7. The most recent interval specified for a device is the interval currently in effect. Keep in mind that when you enter the SET MITIME command for an entire class of devices (such as all DASD or all tape devices), the system monitors *all* members of that class at the new time interval.

For example, assume that the valid range of DASD addresses at your installation is 190 through 400<sup>®</sup>. If you enter **set mitime 250-255 00:59** and at some later point in time enter **set mitime dasd 00:20** the current interval for devices 190 through 400 is 20 seconds.

8. Changes can be made to several device classes with a single command. The command is processed from left to right so that the time interval specified last on the command line for a device class is the interval currently in effect.
9. Only device classes may be repeated on a single command line. If real device numbers are detected along with device classes on a single command line, the command is rejected.
10. The SET MITIME command has no effect on EDEVICES. The SCSI device driver has a separate mechanism and set of timers for managing missing interrupts. The values for those timers are not

configurable. SET MITIME commands that would include EDEVICES in their specification, either with the DASD class or by RDEV, will result in an informational message and will impact only devices in the specification that are not EDEVICES. Device ranges that include EDEVICES will be modified to have the EDEVICES removed, which will result in multiple ranges being displayed on the QUERY MITIME command.

Valid examples of multiple requests:

```
set mitime dasd 00:30 tape off ur 01:00 misc
1:40
set mitime tape 01:50 dasd 02:35 misc 02:30 tape 00:50
set mitime dasd 00:00 ur 05:00 misc 1:40 dasd off
```

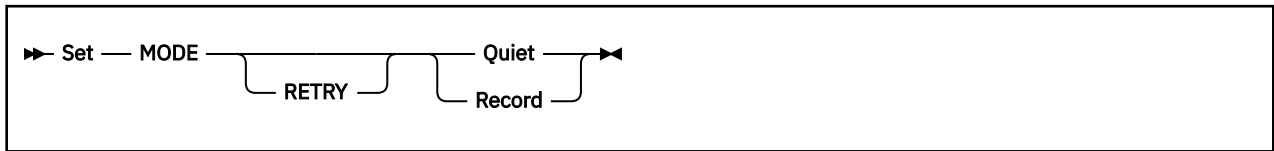
Nonvalid examples:

```
set mitime dasd 00:30 tape off ur 01:00 190
01:40
set mitime 225-260 02:35 tape 00:50
set mitime dasd 01:00 ur 05:00 255 00:30 dasd off
```

## Messages

- HCP026E Operand missing or invalid
- HCP287I One or more EDEVICES exist that match the specification. SET MITIME will be ignored for all EDEVICES.

## SET MODE



### Authorization

Privilege Class: A, F

### Purpose

Use SET MODE to set the error-recording mode for system-recovery machine checks (soft errors).

**Note:** When the hardware recovers from an error without software assistance, it issues a system recovery machine check interrupt to report the incident.

### Operands

#### RETRY

is an optional keyword provided for compatibility with earlier systems.

#### Quiet

turns off error recording of system recovery machine checks.

#### Record

turns on error recording of system recovery machine checks. The next 12 system recovery machine checks are recorded. After that, the system automatically returns to QUIET mode and you receive a message informing you of that condition.

When a warning machine check occurs, a message is issued and further warning machine checks are disabled. The system then runs disabled for further warning machine checks until the SET MODE RECORD command is issued.

### Usage Notes

1. In addition to turning on and off system recovery machine check recording, you can use this command to reenables warning machine checks.

### Responses

Response 1:

```
SYSTEM RECOVERY {QUIET } MODE ACCEPTED
                 {RECORD}
```

indicates that the command has been accepted.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP6005E Option *option* is not supported by z/VM.

## SET MONDATA



### Authorization

Privilege Class: C

### Purpose

Use SET MONDATA to control whether display device input and output data appear in the monitor records.

### Operands

#### ON

indicates that input from, and output to, user display devices is to be included with the terminal event records.

#### OFF

indicates that input from, and output to, user display devices is not to be included with the terminal event records.

### Usage Notes

1. The MONDATA option is set to OFF when the system is IPLed and is not changed unless a user issues SET MONDATA ON.
2. To obtain the setting of the MONDATA option, enter the QUERY MONDATA command.
3. It is the responsibility of the installation to protect the security of the data if MONDATA is set ON.

### Responses

Response 1:

```
Command complete
```

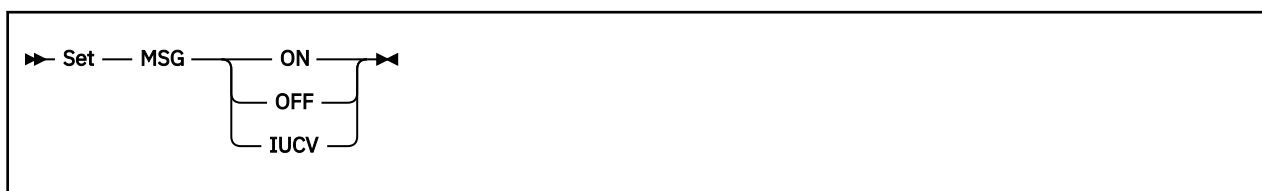
is issued when the command completes successfully.

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid

## SET MSG

---



### Authorization

Privilege Class: G

### Purpose

Use SET MSG to control whether the MESSAGE command messages are to be received at your display. The initial setting for MSG is installation dependent and is defined in the system configuration file.

### Operands

#### ON

#### OFF

controls the display of messages. If you specify ON, the messages are displayed. If you specify OFF, no messages are displayed.

#### IUCV

specifies that messages are to be passed to the virtual machine through IUCV if a connection to the message system service exists. If messages are passed through IUCV, they are not placed in the virtual console spool file. If no IUCV connection exists, messages are handled as if SET MSG ON had been issued.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid



## SET MSGFACIL

---

```
▶▶ Set — MSGFacil — OFF ▶▶
```

### Authorization

Privilege Class: G

### Purpose

Use the SET MSGFACIL OFF command to remove all message facility capability from the issuing virtual machine. All virtual message processors and message devices are detached from the virtual machine.

### Operands

#### OFF

removes all virtual message capability from your virtual machine

### Usage Notes

1. To define a virtual message processor and associated virtual message devices, use the DEFINE MSGPROC command.
2. To display information about virtual message processors in your I/O configuration, use the QUERY VIRTUAL MSGPROC command.
3. To display information about the virtual message devices within your I/O configuration, use the QUERY VIRTUAL MSGDEVICES command.
4. To restart a virtual message processor that is within your I/O configuration, use the RESTART MSGPROC command.
5. To detach a virtual message processor and associated virtual message devices, use the DETACH MSGPROC command.

### Responses

Response 1:

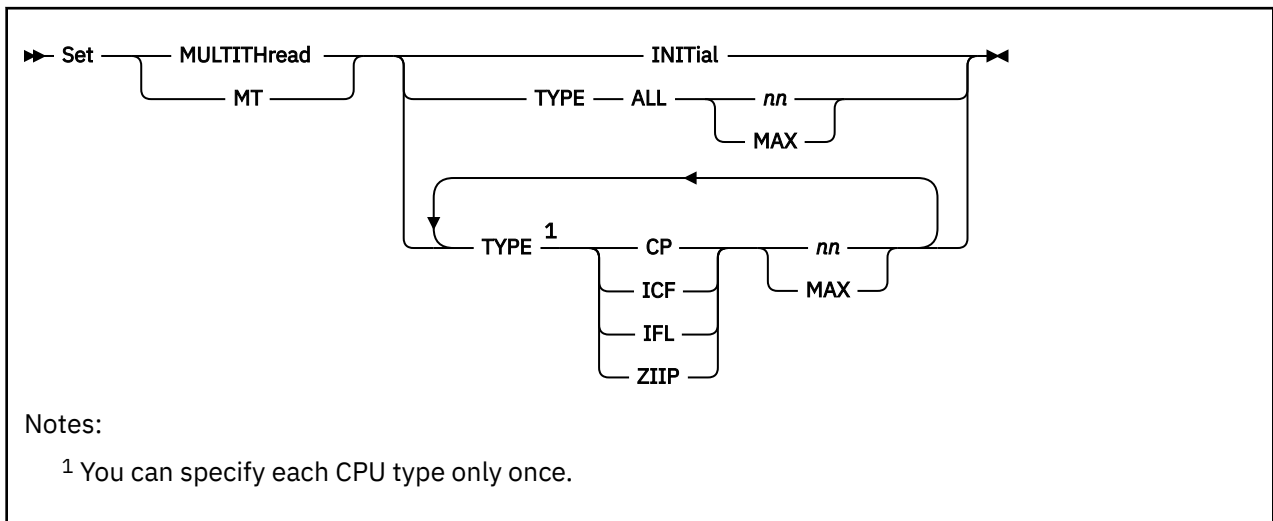
To remove all virtual message facility capability from your virtual machine, enter the following:

```
set msgf off
Command complete
Ready;
```

### Messages

- HCP002E Invalid operand – *operand*
- HCP003E Invalid option – command contains extra option(s) starting with *option*
- HCP003E Invalid option – *option*

## SET MULTITHREAD



### Authorization

Privilege Class: A

### Purpose

Use SET MULTITHREAD to change the multithreading (MT) characteristics of the system. The command allows the activated thread count per CPU type to be changed if multithreading is enabled.

When the result of the command changes the number of activated threads of a CPU type, then logical processors might be activated or deactivated to match the new setting.

### Operands

#### INITIAL

This operand requests that the configuration be restored to the values specified on the MULTITHREADING statement in the system configuration file at the time the system was IPLed. If it is specified, no other operands are allowed.

#### TYPE ALL {nn/MAX}

specifies the CPU type and the requested number of threads for that type. If ALL is specified, no individual CPU types may be specified. The value specified by this operand tells z/VM the number of threads to activate on each CPU type. The value *nn* must be a decimal number 1 - 32 and cannot be greater than the effective maximum threads value. If MAX is specified the activated thread count will be set to the effective maximum threads value. Refer to usage note [2](#) for an explanation of the effective maximum threads value, and usage note [4](#) for an explanation of how it influences the activated thread count.

When ALL is not specified and some of the individual CPU types are not specified, then the unspecified CPU types retain their prior values.

### Usage Notes

1. The activated thread count per CPU type can be changed only if multithreading is enabled.
2. The effective maximum threads value is the maximum number of threads any CPU type can use given the MAX\_THREADS setting requested on the MULTITHREADING statement, hardware limits and software limits. This value is displayed in the Activated Threads column of the MAX\_THREADS row of the QUERY MULTITHREAD command output. For an explanation of the MAX\_THREADS setting, see the [“QUERY MULTITHREAD”](#) on page 1026 command.

3. The number of online logical processors per core is determined by the activated thread count for the corresponding CPU type.
4. The actual activated thread count for a CPU type might be less than what was requested if the hardware or z/VM has a limit less than the effective maximum threads value. This level of z/VM supports up to 2 threads for IFL and 1 thread for all other CPU types.
5. For each CPU type, the resulting activated threads number will be the least of the following:
  - The effective maximum threads value
  - The requested threads value for that CPU type as specified on this command
  - The number of threads per core supported for that CPU type on the hardware or the configuration where z/VM is running
  - The number of threads for that CPU type supported by z/VM
6. The SET MULTITHREAD command is an asynchronous command. The initial synchronous stage of processing parses the command and determines whether or not it is valid. If valid, and it is not necessary to activate or deactivate processors, a "Command complete" response is issued. If it is necessary to activate or deactivate processors then a "Command accepted" response will be issued and control will be returned to the caller after scheduling the asynchronous processing to process the requested transition.
7. During the asynchronous processing step informational messages are displayed:
  - HCP6514I and HCP6517I which indicate the beginning and end of the asynchronous processing are sent to the command issuer and the primary system operator if not the issuer of the SET MULTITHREAD command.
  - HCP6515I and HCP6516I identify each of the processors that have been activated or deactivated as a result of the SET MULTITHREAD command. The primary system operator is also notified if not the issuer of the SET MULTITHREAD command.

These messages are generated after completion of the synchronous portion of the command, they are not treated as command responses. Therefore they will not be returned through interfaces like the AT or FOR command, Diagnose 8, or \*ASYNCMD.
8. Any additional SET MULTITHREAD commands will be rejected until the asynchronous processing has completed. The QUERY MULTITHREAD command can be used to determine when the asynchronous portion of the SET MULTITHREAD command processing has completed. The rejection will be indicated by message HCP6513I and corresponding return code.
9. The QUERY MULTITHREAD INITIAL command can be used to determine what the initial multithreading configuration settings were which would be restored by SET MULTITHREAD INITIAL.

## Responses

Response 1:

```
Command complete
```

Response 2:

```
Command accepted
```

After this response, asynchronous processing will complete the request. During that processing the following informational messages will be issued:

HCP6514I Processor reconfiguration for a multithreading configuration change has started.

HCP6517I Processor reconfiguration for a multithreading configuration change has completed.

One or more of the following information messages may be displayed between HCP6514I and HCP6517I when the SET MULTITHREAD command causes one or more CPU types with online cores to change activated thread counts.

HCP6515I Processor nnnn online due to thread activation

HCP6516I Processor *nnnn* offline due to thread deactivation

### Examples

1. To change the system to run single-threaded, use the following command:

```
SET MULTITHREAD TYPE ALL 1
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP1001E An operand is missing for *option*
- HCP1321E SET MULTITHREAD is not valid because multithreading is not enabled.
- HCP6509I The requested number *nn* of the operand **{ALL|CP|IFL|ICF|ZIIP}** in the SET MULTITHREAD command is larger than the hardware-supported number *hh* or the CP-supported number *cc*, and has been changed to *mm*.
- HCP6512E The requested number *nn* of the operand *{ALL|CP|IFL|ICF|ZIIP}* is larger than the effective maximum threads value *cc*
- HCP6513E *command1* not allowed when a multithreading configuration change is already in progress.
- HCP6514I Processor reconfiguration for a multithreading configuration change has started.
- HCP6515I Processor *nnnn* online due to thread activation
- HCP6516I Processor *nnnn* offline due to thread deactivation
- HCP6517I Processor reconfiguration for a multithreading configuration change has completed.
- HCP6704E Missing token at end of line
- HCP6706E Invalid number - *number*
- HCP9152E Core *mmmm* is being varied offline because processor *nn* is not responsive during thread activation.
- HCP9152E Core *mmmm* is being varied offline because processor *nn* is not responsive during thread deactivation.

## SET NEW\_DEVICES

►► Set — NEW\_DEVICES — INITialize\_when\_added — ON — OFF —

### Authorization

Privilege Class: B

### Purpose

Use SET NEW\_DEVICES to change the way CP handles new real devices as they are added to a running system.

### Operands

#### INITialize\_when\_added ON

tells CP to automatically create a real device control block (RDEV) and initialize (bring online) the associated I/O device when you add a new device to the system.

#### INITialize\_when\_added OFF

tells CP to create a real device control block (RDEV) for a new I/O device, but not to initialize (bring online) that device when your system receives an I/O machine check (IOMCK) for adding a new device to the system. To bring the device online, use the CP VARY (Real Device) command (see [“VARY \(Real Device\)”](#) on page 2063).

### Usage Notes

1. You can also tell CP how to handle new devices using the FEATURES statement in your system configuration file. The FEATURES statement allows you to specify ENABLE (or DISABLE) NEW\_DEVICES\_INITIALIZED\_WHEN\_ADDED. For more information about the [FEATURES Statement](#), see *z/VM: CP Planning and Administration*.
2. To display whether CP initializes new real devices as they are added to a running system, use the QUERY NEW\_DEVICES command (see [“QUERY NEW\\_DEVICES”](#) on page 1033).
3. For more information about adding new devices to a running system, see the [FEATURES Statement](#) in *z/VM: CP Planning and Administration*.

### Responses

Response 1:

To have your devices initialized when they are added to the configuration, enter the following SET NEW\_DEVICES command:

```
set new_devices init on
Command accepted
Ready;
```

Response 2:

To prevent your devices from being initialized when they are added to the configuration, enter the following SET NEW\_DEVICES command:

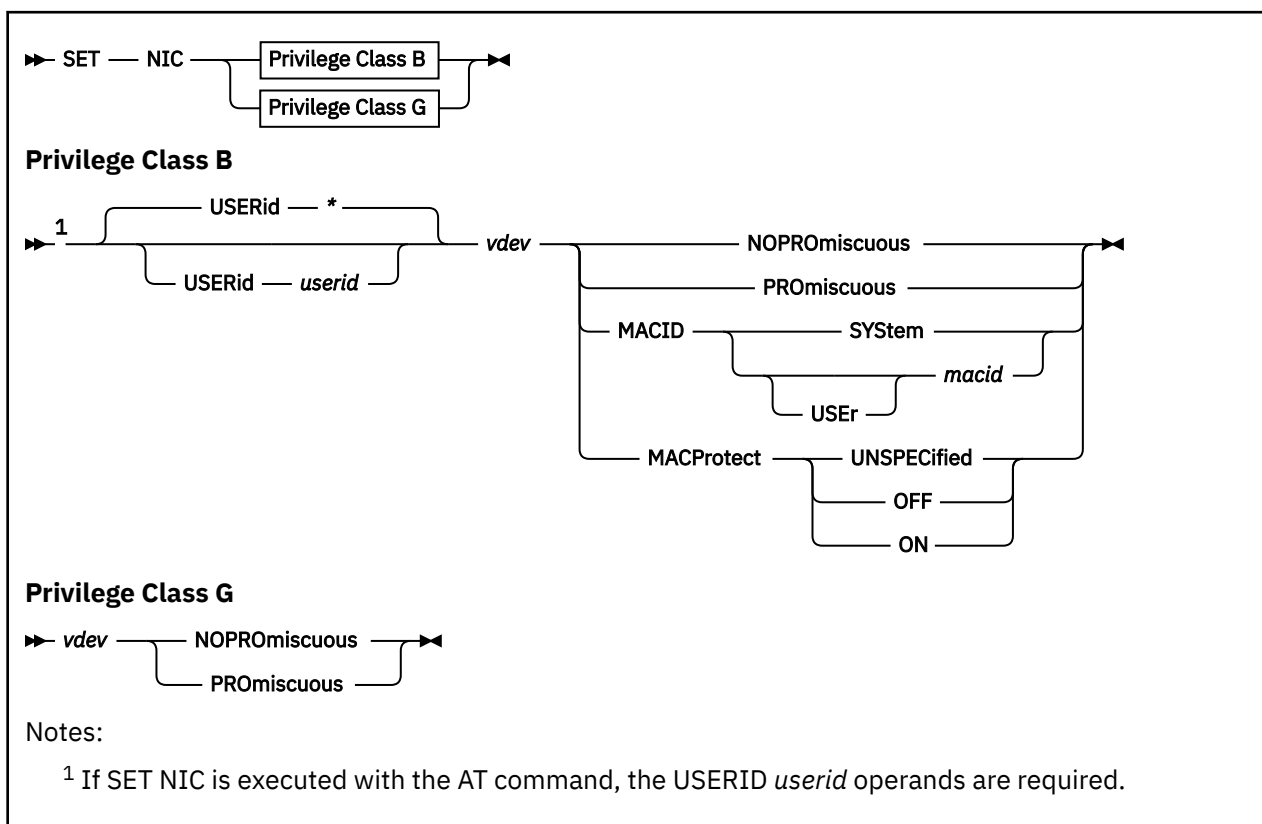
## SET NEW\_DEVICES

```
set new_devices init off  
Command accepted  
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP0003E Invalid option - command contains extra option(s) starting with *option*
- HCP6704E Missing token at end of line

## SET NIC



### Authorization

Privilege Class: B, G

### Purpose

Use SET NIC to perform various settings for a specific network device.

A class B user can:

- Toggle promiscuous mode ON or OFF for a specified user ID's virtual network data device. In order for the guest to participate in promiscuous mode, the NIC must be authorized for promiscuous mode and coupled to a guest LAN or virtual switch.
- Specify the mechanism to be used when assigning a MAC address to a real or simulated network data device.
- Enable or disable MAC address protection for a specific network data device. Enabling MAC address protection ensures only the MAC address assigned by z/VM or the system administrator is used when a guest establishes a layer 2 (ETHERNET) network connection.

A class G user can only toggle promiscuous mode ON and OFF for a particular virtual network data device.

### Operands

**USERid *userid***

**USERid \***

is the user ID of the virtual machine for which you want to set NIC settings. If omitted or specified as an asterisk (\*), CP uses the user ID of the virtual machine issuing the command. The *userid* specified must be logged on.

**vdev**

is the virtual device number to be modified. Typically, this is required to be the device number of what is the data device (or what will be the data device) for the NIC.

**NOPROMiscuous****PROMiscuous**

specifies if promiscuous mode is to be toggled ON or OFF. Current Linux distributions enable or disable Promiscuous mode as part of the device driver alleviating the need for this SET command. Authorization to enable Promiscuous mode is still required using SET VSWITCH.

**MACID SYSTEM****MACID [USER] macid**

specifies the mechanism to be used when assigning a MAC address to a real or simulated network data device.

Specify SYSTEM to have a locally administered MAC address automatically assigned to this device number. The first 3 bytes (6 hexadecimal digits) of the MAC address will be the MACPREFIX value set on the VMLAN system configuration statement and the last 3 bytes will be the next available number in the MACIDRANGE SYSTEM range specified on the VMLAN statement.

Specify USER *macid* to have a static MAC address assigned. The *macid* value must be a unique 3-byte (6 hexadecimal digits) identifier. It will be appended to the system MACPREFIX or USERPREFIX value set on the VMLAN system configuration statement to form a unique user-defined locally administered MAC address for this device.

When MACPREFIX and USERPREFIX are set to the identical value, the value specified for *macid* must be in the USER subset of the MACIDRANGE SYSTEM range.

When MACPREFIX and USERPREFIX are set to different values, the MACIDRANGE settings are ignored. This allows the entire range of MACIDs (000001 - FFFFFFF) to be used by either the system or the user when making MAC address assignments.

Use SET NIC with the SYSTEM operand to remove a previously assigned USER MACID.

**MACProtect UNSPECified|ON|OFF**

turns MAC address protection on or off for the specified network device number. Turning MAC address protection on prevents a guest from using a MAC address that is not assigned to the user's network device. MAC address protection is valid only for simulated network devices when coupled to an ETHERNET virtual switch or guest LAN. MAC address protection is not supported for HiperSockets Bridge Capable guest ports that are being bridged by this virtual switch.

There are three levels of inheritance used when determining the MAC address protection level for a MAC address assigned to a network device. The highest is the system level, followed by the virtual switch or guest LAN level. The lowest level is the protection set for a specific network data device.

The MAC address protection level assigned is determined first by the device, then virtual switch or guest LAN and lastly the system level. If MAC address protection is set at the device level using the SET NIC CP command, then its current setting is used. When the device is set to UNSPECIFIED, it will inherit the MACPROTECT setting of the virtual switch. When the virtual switch MACPROTECT setting is set to UNSPECIFIED, the SYSTEM's MAC address protection specified by the SET VMLAN CP command or VMLAN configuration statement is assigned.

**UNSPECified**

allows the MAC address protection to be determined by the next higher level. UNSPECIFIED is the default setting for a network device.

**OFF**

will set the device MAC address protection level to unrestricted or off. The MAC address assigned to this device is not protected. A guest can override the assigned MAC address when establishing a network connection.

**ON**

will set the device MAC address protection level to restricted or on. The MAC address assigned to this device is protected and cannot be overridden by the guest when establishing a network connection.



## Usage Notes

1. The virtual device number (*vdev*) does not have to exist to set a MACID or the MACPROTECT option. In this case, the option will be preassigned and will be inherited when the virtual NIC is created comprising the device number.
2. A SET NIC *vdev* MACID command has no effect when a real device is attached as *vdev*, unless the device driver uses VM MAC address support. That is, the device driver must make a call to subcode X'30' of Diagnose X'26C' to retrieve a MAC address from CP and use that MAC address on link initialization.  
  
The z/VM TCP/IP stack's OSD device driver provides this support for a link defined to use the ETHERNET transport type. The link must be restarted in order to use the new MAC address. Until the link is restarted, a QUERY VIRTUAL OSA or QUERY VIRTUAL NIC command issued for the device shows a status of "Pending" for the newly assigned MAC address.
3. A SET NIC *vdev* MACID command sets the MAC ID for a virtual device. For the z/VM OSD device driver, the link must be restarted in order to use the newly assigned MAC address, unless it is SET before the link is first initialized..
4. For the z/VM TCP/IP stack, the *network data device number* is the first device number in a network adapter triplet. That is, a DEFINE NIC 501 creates an adapter with devices 501, 502, and 503. 501 is the data device number and should be used in the SET NIC command. (This information applies to z/VM stacks with code from release z/VM 5.4.0 and later.)

## Responses

Response 1:

```
Promiscuous Mode has been enabled for device vdev
```

Response 2:

```
Promiscuous Mode has been disabled for device vdev
```

Response 3:

```
MAC address macaddr has been assigned to device vdev
```

Response 4:

```
A MAC address is automatically assigned to device vdev
```

are the normal responses from this command.

## Messages

- HCP006E Invalid device type - *vdev*.
- HCP0045E *userid* not logged on
- HCP022E A virtual device number was not supplied or it was invalid
- HCP026E Operand missing or invalid
- HCP040E Device *vdev* does not exist.
- HCP2795E Promiscuous Mode can not be used with a HiperSockets device.
- HCP2795E Promiscuous Mode must be used on a data device
- HCP2850E MACID is not set for *vdev*. MAC address *macaddr* is already in use [by *user ID*]
- HCP2850E MACID is not set for *vdev*. Suffix for MAC address *macaddr* is not within USER MACIDRANGE.
- HCP2850E MACID is not set for *vdev*. SSI cluster mode not stable.

## SET NIC

- HCP2850E MACID is not set for *vdev*. Guest relocation active.
- HCP3022E SET NIC not performed; MACPROTECT is not valid for the IP transport type.
- HCP6704E Missing token at end of line

## SET NOPDATA

---



### Authorization

Privilege Class: G

### Purpose

Use SET NOPDATA to control the simulation of NOP CCWs in your virtual printer and punch channel programs.

### Operands

#### ON

#### OFF

controls the simulation of NOP CCWs. If you specify ON and if you specify more than one byte of data, CP attempts to transfer the data addressed by a NOP CCW to the spool file. If you specify OFF, CP simulates the NOP the same way a real device and channel would. The initial setting for a virtual machine is OFF.

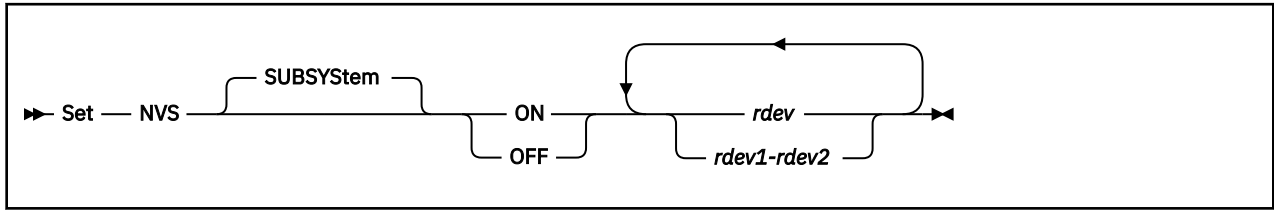
### Usage Notes

1. This command allows programs that need to embed control information in CP spool files (such as RSCS) to operate under z/VM.
2. Do not use data chaining with NOP CCWs.

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP263E Too many operands were supplied for this command.

## SET NVS



### Authorization

Privilege Class: B

### Purpose

Use SET NVS to make nonvolatile storage (NVS) available or unavailable within a DASD subsystem. This command is valid for cache storage subsystems that contain nonvolatile storage.

### Operands

#### SUBSYSstem

is an optional keyword that specifies that the entire storage subsystem is affected by this command, regardless of which device within the subsystem receives the order. This is the default (and only) value.

#### ON

indicates that the nonvolatile storage is to be made available to the subsystem.

#### OFF

indicates that the nonvolatile storage is to be made unavailable to the subsystem.

#### rdev

#### rdev1-rdev2

is the real device number, a list of real device numbers, or a range of real device numbers of devices in the subsystem.

### Usage Notes

1. SET NVS is valid only for cache storage subsystems that have nonvolatile storage installed. An error message is generated for storage subsystems that do not have nonvolatile storage installed.
2. The caching function is always enabled on RAID (Redundant Array of Independent Disks) technology DASD such as an IBM DASD subsystem. For these types of DASD, the device accepts but ignores this command; the caching function cannot be modified by the user. The caching function for these types will always appear as ON in related query commands.
3. Certain error messages may result if the subsystem is not in the correct mode to accept the SET NVS command. See error message HCP296E for more information.
4. If DASD fastwrite has previously been activated for any device, the DASD fastwrite function becomes operational when nonvolatile storage is made available for subsystem use.
5. Making nonvolatile storage unavailable for subsystem use causes a destage of all modified DASD fastwrite data in the subsystem.
6. This command may produce delayed responses. For more information see [“Delayed Responses to CP Commands”](#) on page 7.

### Responses

Response 1:

**Immediate Response:** If you enter the SET NVS ON command and the NVS is made available to the subsystem immediately, the following response is displayed. This response is repeated for each *unique subsystem* identified by the devices specified on the command.

```
rdev Nonvolatile storage has been activated.
```

**rdev**

is the real device number of one of the specified devices in a subsystem for which NVS has been activated.

**Delayed Response:** If the NVS is not made available to the subsystem immediately, the following response is displayed:

```
Command started: SET NVS ON rdev.
```

When the NVS is made available to the subsystem, the following responses are displayed:

```
rdev Nonvolatile storage has been activated.
```

Under certain circumstances, it is possible that a cache storage subsystem does not notify CP when it has made the NVS available to the subsystem. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing is ended after generating the following response:

```
Command results lost: SET NVS ON rdev.
```

In this case, the QUERY NVS command can be issued to determine the state of the nonvolatile storage.

If errors occur during the asynchronous processing of the command that prevent the successful completion of the command, the following response is displayed:

```
Command failed: SET NVS ON rdev.
```

**rdev**

is the real device number of one of the specified devices in a subsystem for which the command has failed.

**Response to System Operator:** If the originator of the command is not the system operator, the following response is issued to the system operator when the NVS has been made available to the subsystem. This response is also repeated for each *unique subsystem* identified by the devices specified on the command line.

```
rdev Nonvolatile storage was activated by userid.
```

**rdev**

is the real device number of one of the specified devices in a subsystem for which the NVS has been activated.

**userid**

is the ID of the user who issued the command.

Response 2:

**Immediate Response:** If you enter the SET NVS OFF command and the NVS is made unavailable to the subsystem immediately, the following response is displayed. This response is repeated for each *unique subsystem* identified by the devices specified on the command

```
rdev Nonvolatile storage has been deactivated.
```

**rdev**

is the real device number of one of the specified devices in a subsystem for which the NVS has been deactivated.

**Delayed Response:** If the NVS is not made unavailable to the subsystem immediately, the following response is displayed:

```
Command started: SET NVS OFF rdev.
```

When the NVS is made unavailable to the subsystem, the following responses are displayed:

```
rdev Nonvolatile storage has been deactivated.
```

Under certain circumstances, it is possible that a cache storage subsystem does not notify CP when it has made the NVS unavailable to the subsystem. When this occurs, the appropriate command response cannot be generated. Under these circumstances, command processing is ended after generating the following response:

```
Command results lost: SET NVS OFF rdev.
```

In this case, the QUERY NVS command can be issued to determine the state of the nonvolatile storage.

If errors occur during the asynchronous processing of the command that prevent the successful completion of the command, the following response is displayed:

```
Command failed: SET NVS OFF rdev.
```

#### **rdev**

is the real device number of one of the specified devices in a subsystem for which the command has failed.

**Response to System Operator:** If the originator of the command is not the system operator, the following response is issued to the system operator when the NVS has been made unavailable to the subsystem. This response is also repeated for each **unique subsystem** identified by the devices specified on the command line.

```
rdev Nonvolatile storage was deactivated by userid.
```

#### **rdev**

is the real device number of one of the specified devices in a subsystem for which the NVS has been deactivated.

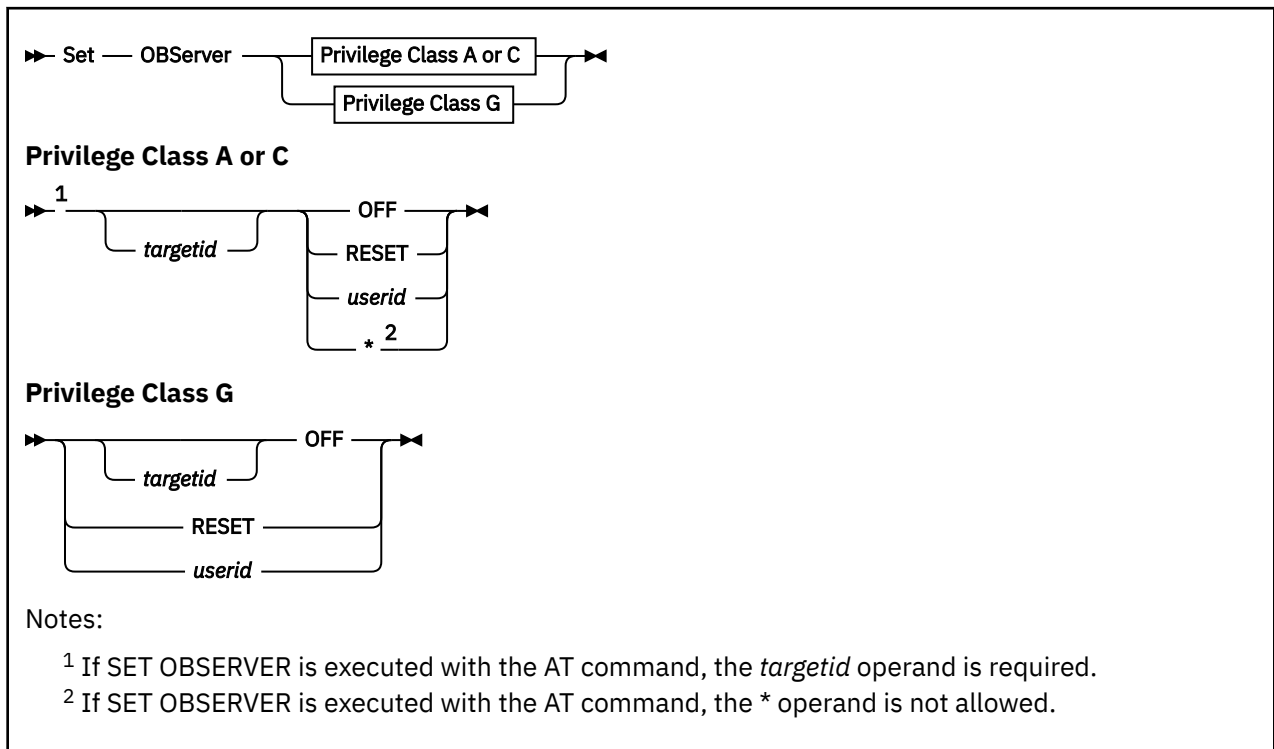
#### **userid**

is the ID of the user who issued the command.

## Messages

- HCP006E Invalid device type - {rdev/vdev/ldev}
- HCP009E Invalid range - range
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {rdev/vdev/ldev} does not exist
- HCP046E type rdev offline
- HCP319E Error processing subsystem command - cmd for DASD rdev
- HCP332E Invalid control unit type - rdev
- HCP1109E The command command cannot be processed for device rdev because the device is the secondary in a duplex pair.

## SET OBSERVER



### Authorization

Privilege Class: A, C, G

### Purpose

Use SET OBSERVER to change the observer setting for a virtual machine. The observer is a virtual machine that receives the line-mode output produced on another virtual machine's console. If you are a class G user, you can change the observer setting for your own virtual machine. If you are a class A or C user, you can change the observer setting for any virtual machine.

### Operands

#### *targetid*

is the user ID of the virtual machine whose observer status is to be set or reset. If you are a class A or C user and you do not specify *targetid*, CP sets or resets the secondary user for your user ID. If you are a class G user, you can stop being the observer for *targetid* by specifying *targetid* OFF.

#### *userid*

is the user ID of the observer virtual machine.

#### RESET

sets the observer for *targetid* or the command issuer to the user ID specified on the CONSOLE statement in the user's directory entry.

#### OFF

removes the user from observation.

#### \*

sets the observer to the virtual machine that issues the command.

## Usage Notes

1. If observer support is used with large console messages that are close to the free storage limit of X'1FD' doublewords, it is possible that the message could be truncated at the end in order to add the header at the beginning which indicates the user ID from which the message originated.
2. A virtual machine cannot have both an observer and a secondary user defined by any combination of the SET OBSERVER and SET SECUSER commands and the CONSOLE directory statement.
3. If the designated observer is currently unable to function in that capacity (for example, not logged on, or disconnected with no connection to \*MSG or \*MSGALL), the setting is nevertheless accepted by CP, and the relationship is established when the specified user ID meets the necessary criteria.
4. In an SSI cluster, the observer and the target (observed) user can be logged on to different member systems.

If either user is a multiconfiguration virtual machine, the observer will function in that capacity only when it is local (logged on to the same member as the observed user). If the observed user is a multiconfiguration virtual machine, the observer must be logged on to the member where the SET OBSERVER command was issued.

## Examples

Example 1:

USERA (a general user) designates USERB (another general user) as its observer by issuing the following command:

```
set observer userb
```

USERB can stop observing USERA by issuing the following command:

```
set observer usera off
```

If USERC is the observer for USERD, USERD can remove its observer by issuing the following command:

```
set observer off
```

Example 2:

USERC (a class A or class C user) issues the following command to assign USERD as the observer for USERB:

```
set observer userb userd
```

USERD receives the following message:

```
HCP6768I Observation of USERB initiated for you by USERC
```

At the same time, USERB receives one of the following messages:

```
HCP6768I Your observation by USERD set by USERC
```

-or-

```
HCP6769I Your observation terminated by USERC
```

Example 3:

USERC can remove the observer for USERB by issuing the following command:

```
set observer userb off
```

This causes the user ID that was the secondary user (USERD) to receive the message:



```
HCP6769I Observation of USERB terminated for you by USERC
```

#### Example 4:

If the command issuer becomes the observer, the message issued at initiation is:

```
HCP6768I Observation of target initiated
```

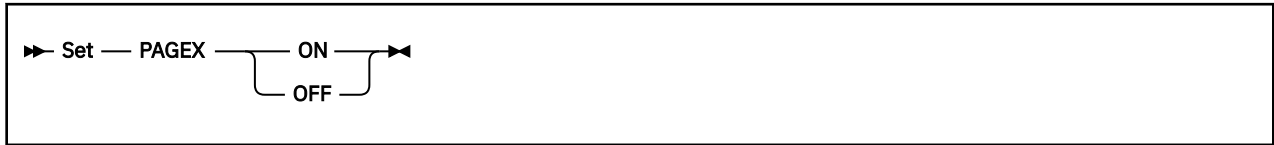
When the secondary user option is terminated, the message issued is:

```
HCP6769I Observation of target terminated
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) beginning with *option*
- HCP007E Invalid userid - *userid*
- HCP017E User *userid* has a SECUSER and cannot be observed
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP053E *userid* not in CP directory
- HCP1149E Error communicating with another SSI member. It cannot be determined if *command* executed remotely.
- HCP6768I Observation of *target* by *observer* is not functional
- HCP6768I Observation of *target* initiated
- HCP6768I Observation of *target* initiated for you by *userid*
- HCP6768I Your observation by *observer* set by *userid*
- HCP6769I Observation of *target* terminated
- HCP6769I Observation of *target* terminated for you by *userid*
- HCP6769I Your observation terminated by *userid*

## SET PAGEX



### Authorization

Privilege Class: G

### Purpose

Use SET PAGEX to turn on or off the pseudo-page-fault facility for your virtual machine.

When the facility is on, your virtual machine can dispatch a task while waiting for CP to satisfy a page-in request for another task. When the facility is off, your virtual machine must wait until CP has satisfied a pending page-in request before dispatching another task.

**Note:** In general, your virtual machine guest operating system issues this command without your intervention. The guest operating system will do this explicitly using DIAGNOSE code X'08', which implies that the guest already knows (from STIDP) that it is running in a virtual machine.

### Operands

**ON**

**OFF**

controls whether the pseudo-page-fault facility is activated or deactivated for your virtual machine. ON activates the facility. OFF deactivates it.

### Usage Notes

1. In general, do not enter this command arbitrarily. If you enter the command while a guest operating system is running, you may cause the system to be stopped or to end abnormally.
2. The pseudo-page-fault facility is an interface consisting of a special CP program interrupt code (X'14'). The guest operating system must have this special program interrupt handler to switch to another task while CP resolves the host page fault that triggered the PAGEX interruption. When the host page fault has been resolved, CP presents a second interruption to the virtual machine for each task that previously received a PAGEX interruption for an address in the page where the fault occurred. See [“Pseudo-Page-Fault Interruption” on page 1595](#) below for details.

If the pseudo-page-fault facility is not activated for an operating system that supports the PAGEX interface feature, you may cause performance degradation. On the other hand, if the pseudo-page-fault facility is activated for an operating system that does *not* support the handshaking feature, results are unpredictable as the guest program interrupt handler will not recognize the special PAGEX program interrupts.

3. For ESA/XC and z/XC virtual machines, SET PAGEX governs handling of faults to only the host-primary address space. Page-fault handshaking of access register (AR)-specified references, as controlled by the PFAULT macro, is not affected by SET PAGEX in any way, nor is SET PAGEX affected by this support. See [PFAULT Macro -- Page-Fault Handshaking Services in z/VM: CP Programming Services](#) for details on page-fault handshaking for AR-specified references.
4. PAGEX is used by the following IBM operating systems:

VSE/ESA  
GCS

5. If the virtual configuration executes a SIGP Set-Architecture instruction, the pseudo-page-fault facility is reset.
6. SET PAGEX ON may not be issued when the asynchronous page-fault handshaking facility for the host-primary address space is in effect (when the VERSION=2 parameter of the PFAULT TOKEN macro has been specified).

## Pseudo-Page-Fault Interruption

A pseudo-page-fault (PPF) interruption is a program interruption with interruption code X'0014'. There are two types of PPF interruptions, initial and completion. When PPF interruptions are enabled and a guest instruction encounters a CP page fault which can not be resolved immediately, a PPF initial interruption is presented to the virtual CPU. This allows the guest supervisor to suspend the running thread and dispatch another. For each initial interruption, a completion interruption specifying the same fault address is presented to the same virtual CPU when the page fault has been resolved. This signals the guest supervisor to unblock the affected thread.

If the virtual CPU is disabled for PPF interruptions when a page fault occurs, the fault is handled synchronously; that is, execution of the virtual CPU is suspended until the fault is resolved. If the virtual CPU is disabled for PPF interruptions when a previous asynchronous page fault is completed, the completion interruption remains pending until the virtual CPU enables.

PPF interruptions are enabled on a virtual CPU when all of the following are true:

- PAGEX is set ON for the virtual machine.
- The PSW is enabled for I/O interruptions. (PSW bit 6 is one.)

In addition, if the virtual machine is in z/Architecture or z/XC mode, PPFs are presented only for faults on guest real addresses below 2 GB. Faults on higher addresses are handled synchronously, as if PPF interruptions were disabled.

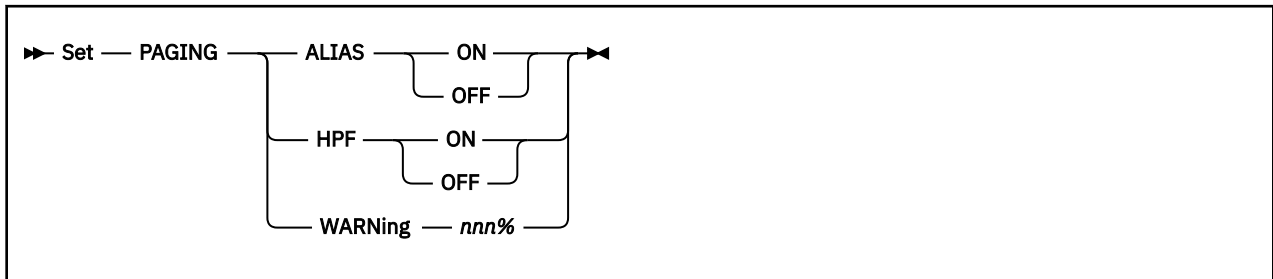
When a PPF interruption is presented, an interruption parameter is stored at guest real locations 144-147 (X'90'-X'93'), as follows:

- Bit 0 is zero for a PPF initial interruption, or one for a PPF completion interruption.
- Bits 1-31 are the guest real starting address of the page on which the fault occurred. (Bits 20-31 are zero.)

## Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP1404E Asynchronous page-fault handshaking for the host-primary space is already in effect. PAGEX cannot be activated.

## SET PAGING



### Authorization

Privilege Class: B

### Purpose

Use the SET PAGING command to modify settings for the paging subsystem.

### Operands

#### ALIAS ON

indicates that system-attached HyperPAV aliases should be used for paging I/O. HyperPAV aliases allow concurrent I/O requests to the same physical volume by using alias devices. Aliases do not have associated storage and can be bound dynamically to base volumes. Using HyperPAV can increase throughput by reducing queuing at the device level.

#### ALIAS OFF

indicates that system-attached HyperPAV aliases should not be used for paging I/O. This is the default behavior, unless aliases are activated on a PAGING or FEATURES statement in the system configuration file (see Usage Note “1” on page 1597), or through a previous SET PAGING command.

#### HPF ON

indicates that transport mode channel program format should be used for paging I/O. With transport mode, less overhead occurs between the channel subsystem and the FICON adapter than with traditional command-mode I/O. This results in higher I/O rates and less CPU overhead.

Specifying SET PAGING HPF ON is not recommended unless all paths to paging devices are High Performance FICON for IBM Z (zHPF) capable.

#### HPF OFF

indicates that command-mode channel program format should be used for paging I/O. This is the default behavior, unless HPF is activated on a PAGING or FEATURES statement in the system configuration file (see Usage Note “1” on page 1597), or through a previous SET PAGING command.

#### WARNING

defines the allocation warning level for paging volumes. The *nnn* value is a decimal integer in the range of 0 to 100. If the number of allocated slots reaches this percentage of total slots on all paging volumes, message HCP401I is issued to alert the system operator.

This warning level is in addition to the default 90% warning message (issued via message HCP401I). Setting the WARNING value to 90% does not result in a second warning message.

This value also serves as the default MAXPAGEFULL setting for a SET STORAGE command to make sure a storage removal does not cause paging volumes to exceed this level of allocation. See “[SET STORAGE](#)” on page 1722 for more information.

## Usage Notes

1. Use the PAGING statement in the system configuration file to establish the paging subsystem's settings at IPL.

For compatibility, the Alias and HPF settings can also be specified on the FEATURES statement in the system configuration file using the PAGING\_ALIAS and PAGING HPF operands. However, it is recommended that you consolidate your paging options on the PAGING statement.

2. Setting a PAGING WARNING value of 0% results in message HCP401I as soon as the system starts paging operations. After paging slots have been allocated, it is unlikely that the total number of paging slots allocated will revert to 0, even if all guests with slots on DASDs are logged off. This is because some CP-pageable pages (system-owned utility address space pages, for example) might remain allocated.

To reduce paging slot allocations, you could consider specifying STORAGE AGELIST KEEPSLOT NO in the system configuration file, or on the SET AGELIST command, to cause paging slots to be unallocated when the corresponding page is read into memory. KEEPSLOT NO can reduce the number of slot allocations on paging volumes. However, it can also increase paging I/O during page-steal operations because every page needs to be written, even those that are unchanged since the last read from DASD.

## Responses

Response 1:

```
PAGING_ALIAS: {ON}          PAGING_HPF= {ON}          WARNING: nnn%
               {OFF}                {OFF}
```

### PAGING\_ALIAS

indicates the state of the HyperPAV alias feature of the paging subsystem.

### PAGING\_HPF

indicates the state of the HPF feature of the paging subsystem.

#### ON

the feature is active

#### OFF

the feature is not active

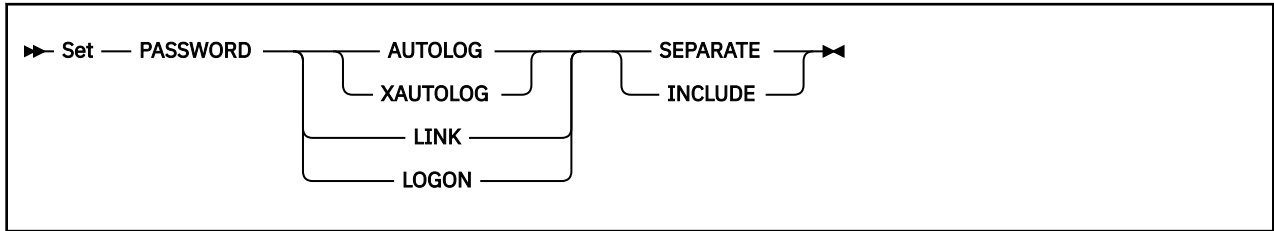
### Warning *nnn%*

specifies the allocation warning level for paging volumes.

## Messages

- HCP003E Invalid option - *{option | command contains extra option(s) starting with option}*
- HCP026E Operand missing or invalid

## SET PASSWORD



### Authorization

Privilege Class: B

### Purpose

Use SET PASSWORD to indicate whether users of commands requiring passwords must enter the passwords separately from the command.

### Operands

**AUTOLOG SEPARATE**  
**XAUTOLOG SEPARATE**  
**AUTOLOG INCLUDE**  
**XAUTOLOG INCLUDE**

sets the system-wide password suppression facility for the AUTOLOG and XAUTOLOG commands. (Specifying either AUTOLOG or XAUTOLOG changes the setting for both commands.) Specify SEPARATE to indicate that passwords cannot be entered as part of the command line. Specify INCLUDE to indicate that passwords *must* be part of the AUTOLOG command line and are permitted as part of the XAUTOLOG command line.

**LINK SEPARATE**  
**LINK INCLUDE**

sets the system-wide password suppression facility for the LINK command. Specify SEPARATE to indicate that passwords cannot be entered as part of the command line. Specify INCLUDE to indicate that passwords are permitted as part of the command line.

**LOGON SEPARATE**  
**LOGON INCLUDE**

sets the system-wide password suppression facility for the LOGON command. Specify SEPARATE to indicate that passwords cannot be entered as part of the command line. Specify INCLUDE to indicate that passwords are permitted as part of the command line.

### Usage Notes

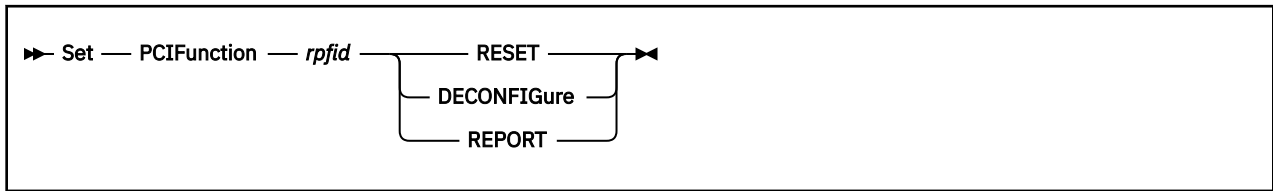
1. For security reasons, the password suppression facility permits your installation to require that issuers of commands that require passwords do not include the password in the command line. Instead, after the user enters the command without the password, the system prompts the user to enter the password, and the input field is masked.
2. The initial (default) system setting for the password suppression facility for the AUTOLOG, XAUTOLOG, LINK, and LOGON commands is the equivalent of SEPARATE. The default system setting can be overridden by the PASSWORDS\_ON\_CMDs operand of the FEATURES system configuration statement. The SET PASSWORD command changes the current setting. Password suppression settings that are not explicitly reset with SET PASSWORD retain their previous values.

3. SET PASSWORD changes the syntax allowed for the specified command, because it indicates whether the password may be included on the command line. Therefore, changing this setting may affect EXECs that use the command.
4. System programmers responsible for z/VM security should examine the usage notes for the `PASSWORDS_ON_CMDs` operand of the [FEATURES Statement](#) in *z/VM: CP Planning and Administration*.

**Messages**

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid

## SET PCIFUNCTION



### Authorization

Privilege Class: B

### Purpose

Use SET PCIFUNCTION to reset or deconfigure a real PCI function or to report an error with the function to the Support Element.

### Operands

#### *rpfid*

is a real PCI function ID to be set. Specify the ID as eight hexadecimal digits. You can omit leading zeros.

#### **RESET**

resets the real PCI function.

#### **DECONFIGure**

removes the real PCI function from the z/VM LPAR configuration.

#### **REPORT**

reports device errors to the Support Element.

### Usage Notes

1. The REPORT operand provides device information for any function type. However, the device information includes the detail information that is displayed by the QUERY PCIF DETAILS command only for NVMe PCI functions that are associated with an EDEVICE.
2. An operation that uses the RESET operand places the PCI function in an error state. After the reset function completes successfully for an NVMe function, the function returns to the operational state and the following message is issued:
  - HCPNVM1164I PCI function *rpcf* recovery was successful.
3. An operation that uses the DECONFIGURE operand for an NVMe function removes the PCI function from the z/VM LPAR configuration. The following operator messages might be issued asynchronously:
  - HCPPCI1164I - PCI Function *rpcf* is no longer available for use by the configuration.
  - HCPPCH1165I - PCI Function *rpcf* has been deleted from the configuration.

Recovery requires that you add the PCI function to the configuration again by using the Support Element or the Hardware Management Console.
4. An operation that uses the RESET or DECONFIGURE operand for an NVMe function places any associated NVMe EDEVICES in an error state. Guests and z/VM might not operate effectively. Recovery requires that you delete and re-create the NVMe EDEVICES.

### Responses

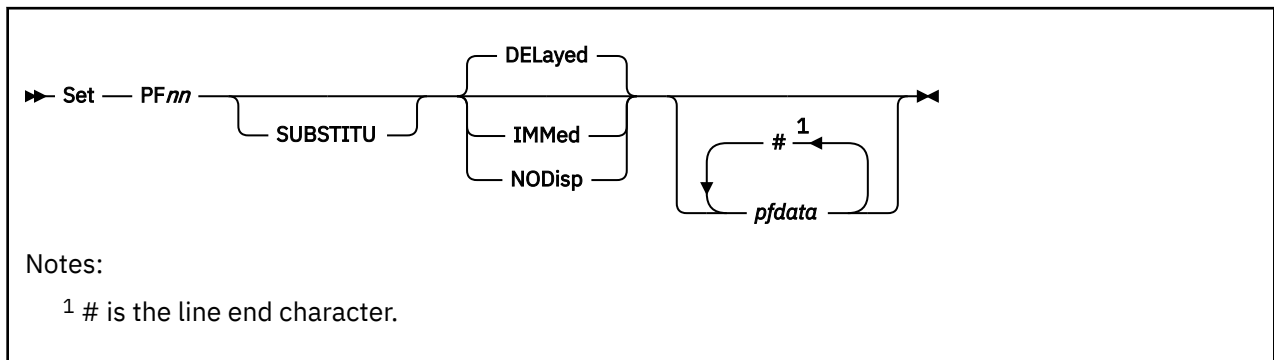
There is no direct response to a successful SET PCIFUNCTION command.



**Messages**

- HCP021E A real PCI function ID was not supplied or it is invalid.
- HCP040E PCI function *rpfid* does not exist
- HCP1003E *rpfid* not found.
- HCP1170E Support Element notification failed for PCI function *rpfid*
- HCP1170E Support Element operation failed for PCI function *rpfid*

## SET PFnn



### Authorization

Privilege Class: G

### Purpose

Use SET PFnn to:

- Assign a program function to a program function key
- Specify when the program function is to be processed.

**Note:** PF keys are part of the terminal display keyboard.

### Operands

#### PFnn

identifies the program function key you are defining; the operand *nn* is a number from 1 (or 01) to 24 that corresponds to a program function key on the display. The program function is a programming capability you create by defining a series of commands you want run. This series of commands runs when you press the appropriate program function key.

#### SUBSTITU

indicates that the PF data may contain one or more substitution place holders.

A substitution place holder is identified by an ampersand followed by an asterisk (&\*) or an ampersand followed by a decimal number comprised of 1 to 8 digits (&n, where n=1,2,3...). The value of the number must be greater than zero. The end of the place holder is delimited by a period, blank, line end, or data end.

Any other ampersand that is not a substitution place holder must be specified within the PF data as two ampersands.

If a single ampersand is detected that is not a valid substitution place holder, the ampersand and what follows to the ending delimiter is translated to a null within the PF data when the PF key is processed.

If SUBSTITU is not specified, no substitution is made.

If SUBSTITU is specified, any &n (for example, &1, &2,...) in PF data is substituted with a token from the input area. The value &1 is substituted with the first token, &2 is substituted with the second token, and so on. &\* is substituted with all the data past the highest token referenced in PF data.

The SUBSTITU operand supports mixed DBCS strings. If you are using DBCS data, SUBSTITU will not look for a substitution character within the DBCS data.

If no corresponding token is found for an &n, it is translated as a null. If a token is not referred to in the PF data, it is ignored.

A token can be delimited by blanks or single quotation marks. If you want single quotation marks imbedded in the token, double the number of quotation marks.

For example, assume you enter **set pf1 substitu msg &1 please attach &2 to me as &3, &4**. If you then enter **operator 135 246 'as soon as possible'** and press the PF1 key, the following appears in your input area:

```
MSG OPERATOR PLEASE ATTACH 135 TO ME AS 246,
AS SOON AS POSSIBLE
```

Specifying a period after a substitution place holder concatenates the substitution place holder with the following data. (The data can be text or another substitution character.) For example, assume you set PF2 by entering:

```
set pf2 substitu immed display Lt&1..&2
```

If you enter:

```
1fd000 90
```

and press PF2, the following command is processed immediately after being displayed:

```
DISPLAY LT1fd000.90
```

CP does not read double-byte characters as tokens.

### **DElayed**

### **IMMed**

### **NODisp**

indicates when the program function is processed. If you specify IMMED, the program function is processed when you press the PF key. If you specify DELAYED, the PF data is displayed in the input area when you press the PF key. The data may then be modified if necessary and the ENTER key pressed when you want the function to be processed. If you specify NODISP, the program function is processed when you press the PF key, but the function is not redisplayed on the terminal. DELAYED is the default.

### ***pfdata***

defines the command or data lines that constitute the program function. If you are entering more than one command line as program function data, use the line end symbol to separate the lines. The default line end symbol is the pound sign (#); you can define your own line end symbol with the TERMINAL LINEND command.

If you use the line end symbol to separate commands that you want run with a designated PF key, you must precede each line end symbol with the logical escape character. The default logical escape character is a double quotation mark symbol (""); you can define your own logical escape character with the TERMINAL ESCAPE command.

If you do not enter any command lines, the SET PFnn command resets a PF key to undefined or resets the IMMED or DELAY status of an existing PF key function to the value you have specified.

Mixed DBCS strings can be used for the data defined for a PF key.

## **Usage Notes**

1. If your keyboard has 24 PF keys, specify the PF key number, *nn*, as a value from 1 (or 01) to 24. If your keyboard has only 12 PF keys, specify *nn* as a value from 1 (or 01) to 12.
2. For DELAYED, if the length of the saved data is longer than the input area, it is truncated to the input area length. The input area length is twice the terminal line size minus 21. The QUERY TERMINAL command returns the LINESIZE field used in this calculation.

## SET PFnn

For IMMED and NODISP, the length of the data is limited to the greater of either the input area length or 240.

**Note:** Remember that this applies to the terminal where you will be pressing the PF key.

3. You can use the line end symbol to separate commands that you want to run with the designated program function key.

For example, suppose your virtual machine has the line editing function set ON, and is using the pound sign (#) as the logical line end symbol.

While in CMS mode, enter **set linedit off**. Then enter **set pf02 immed q rdr:#q pun** or **cp set pf02 immed q rdr:#q pun**. (CP assigns the two QUERY commands as functions of the PF02 key.) Then enter **set linedit on**. (Pressing the PF02 key runs the two QUERY commands.)

Since the entire input line is accepted as valid program function data, do not specify further SET command functions on the SET PFnn command line.

4. On a DELAY PF key, if SUBSTITU is not specified, the cursor is positioned at the end of the data. If SUBSTITU is specified and all of the &ns are substituted with tokens, the cursor is also positioned at the end. If some &ns are not substituted, the cursor is positioned over the first &n that is not substituted.
5. If you press a PF key that has not been set, the following message is displayed:

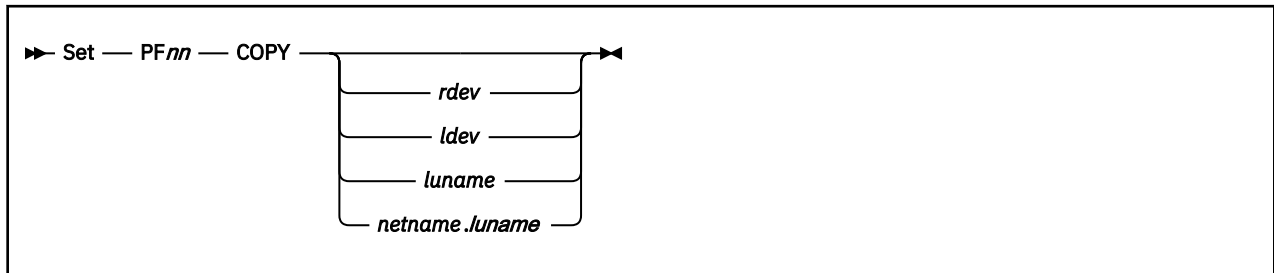
```
PFnn UNDEFINED
```

6. To use the DELAY function on a 3101, you must be in block mode.
7. On a 3101 device, if you press a PF key that has been defined using SUBSTITU, the &ns are always substituted with nulls.

## Messages

- HCP026E Operand missing or invalid

## SET PFnn COPY



### Authorization

Privilege Class: G

### Purpose

Use SET PFnn COPY to copy your display screen to a printer.

### Operands

#### PFnn

identifies the program function key to perform the COPY function for your 3270 or 3290 display.

#### rdev

#### ldev

The operand *rdev* represents the real device number of the display printer to be used; *ldev* is the logical device number of the logical display printer to be used.

#### luname

identifies a printer controlled by the same VTAM service machine that controls the display whose screen is being copied. It is the name from the VTAM definition of the printer as an SNA logical unit. If the *luname* is omitted, it is assumed that the associated control unit supports the local copy function.

#### netname.luname

identifies the network qualified logical unit to be used as the copy device.

### Usage Notes

1. If your keyboard has 24 PF keys, specify the PF key number, *nn*, as a value from 1 (or 01) to 24. If your keyboard has only 12 PF keys, specify *nn* as a value from 1 (or 01) to 12. When you press the PF key, the COPY function produces a printed output of the entire 3270 or 3290 display screen except for the status field, which is replaced by blanks.
2. After you press the PF key, you receive a NOT ACCEPTED message in the status field of your display if any of the following situations occur:
  - The printer is already busy
  - The printer is turned off
  - The printer is operational but has not been varied online or enabled by the system operator
  - The printer is out of paper or is in any other condition that requires intervention
  - The designated device is not a 3286/3287/3288/3289 printer
  - The SET PFnn COPY command format is not valid.

You may include your own identification on the printed output by entering the data into the user input area of the screen before you press the PF key. The identification appears on the last two lines of the printed copy.

## SET PFnn COPY

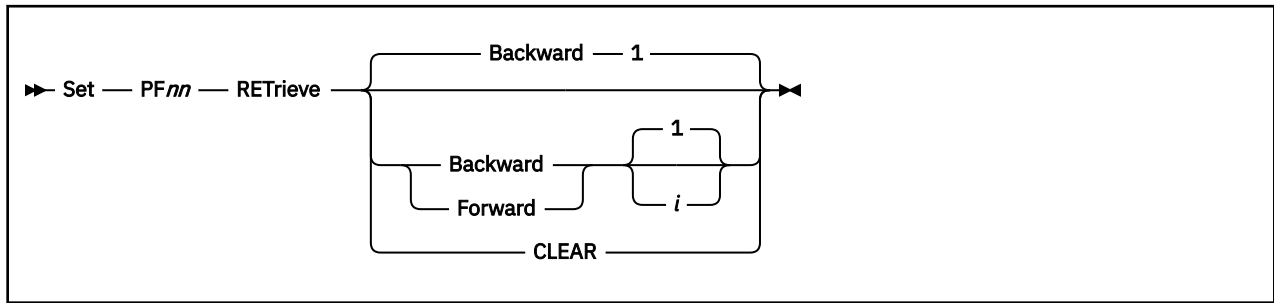
3. To ensure the definition of a PF key for the COPY function, you must start with the keyword COPY, followed by at least one blank, followed by the device identifier (*rdev*, *ldev* or *luname*).
4. The existence of more than one data field (*rdev*, *ldev*, or *luname*) following the keyword COPY identifies the string as PF key data rather than a hard copy request.
5. The following conditions cause the string to be interpreted as PF key data rather than as a hard-copy request:
  - No operand after COPY
  - No *rdev* or *ldev* defined in the range X'0'-X'FFFF'
  - The presence of an operand after *rdev*, *ldev*, or *luname*.
6. SET PFnn COPY is an immediate (nondisplaying) command; its function is processed immediately upon pressing the appropriate program function key. If you insert the keyword DELAY after the PFnn operand, it is ignored; the program function still processes immediately when the PF key is pressed.
7. On all logical display printers, a maximum of 1920 bytes are printed.
8. A copy from a larger display screen to a smaller print buffer produces unpredictable results.
9. The COPY function is not supported on a 3101 device. The 3101 can have an attached printer and the screen can be copied using the PRINT key.
10. If you press a PF key that has not been set, the following message is displayed:

```
PFnn UNDEFINED
```

### Messages

- HCP026E Operand missing or invalid

## SET PFnn RETRIEVE



### Authorization

Privilege Class: G

### Purpose

Use SET PFnn RETRIEVE to assign a RETRIEVE function to a PF key. Use the PF key to retrieve commands and search forward and backward through the RETRIEVE buffers.

### Operands

*nn*

is the number of the program function key for which you are setting the RETRIEVE function.

**Backward *i***

**Forward *i***

indicates the direction that CP skips through the RETRIEVE buffers. When this program function key is pressed, CP searches through the RETRIEVE buffers *i* positions (or 1 if *i* is not specified) and displays the command saved in that RETRIEVE buffer only.

**CLEAR**

causes CP to discard the contents of any existing RETRIEVE buffers. This option also causes CP to define this program function key as if you had entered the command SET PFnn RETRIEVE BACKWARD 1.

### Usage Notes

1. If your keyboard has 24 PF keys, specify the PF key number, *nn*, as a value from 1 (or 01) to 24. If your keyboard has 12 PF keys, specify *nn* as a value from 1 (or 01) to 12.
2. After you define a RETRIEVE program function key or run the SET RETRIEVE command, CP starts saving your input lines. If you enter SET PF11 RETRIEVE BACKWARD 5 and then SET PF12 RETRIEVE FORWARD 3, pressing PF11 will move a pointer to the RETRIEVE buffer five buffers back. If you then press PF12, CP moves the pointer up three buffers to the second one back. Once you press ENTER to enter what was in the RETRIEVE buffer you selected, CP resets the pointer back to the most recent buffer.
3. As a default, CP keeps track of seven input lines. See the SET RETRIEVE command for modifying the number of input lines that are saved and for the effects that SET PFnn can have on the SET RETRIEVE command. Any input line that is an exact duplicate of the previous input line is not saved. For security reasons, input lines that are not displayed on the display device, such as passwords, are not saved.
4. If the saved line is longer than the input area, the line is truncated to the input area length.
5. Do not use any operands before RETRIEVE. If you do so, CP interprets the RETRIEVE as PF data rather than the RETRIEVE function.
6. To use the RETRIEVE function on a 3101, you must be in block mode.

7. If you press a PF key that has not been set, the following message is displayed:

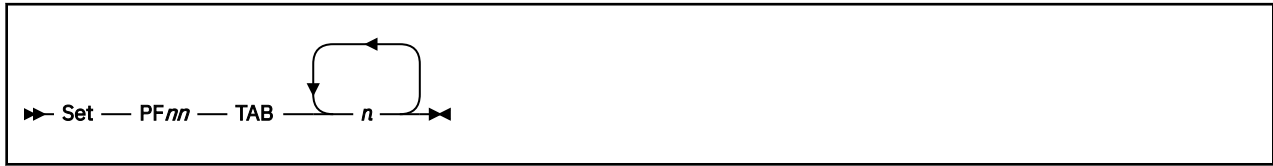
```
PFnn UNDEFINED
```

### Messages

- HCP026E Operand missing or invalid
- HCP6892E Invalid number of buffer positions - *nnn*



## SET PFnn TAB



### Authorization

Privilege Class: G

### Purpose

Use SET PFnn TAB to set values for the TAB function.

### Operands

#### PFnn

is the program function key to perform the TAB function for your virtual machine.

#### n

identifies the tab settings that you want to associate with the PF key.

### Usage Notes

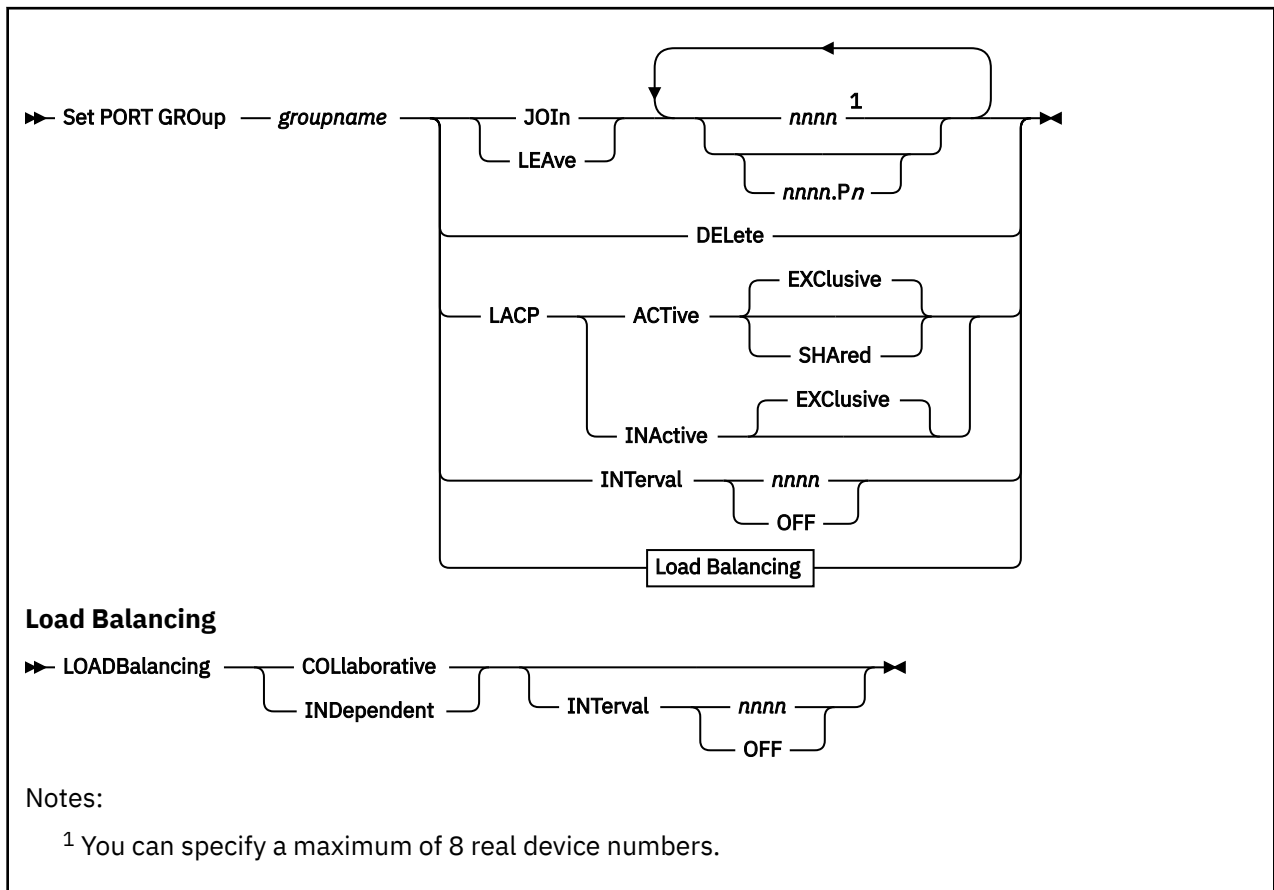
1. If your keyboard has 24 PF keys, specify the PF key number, *nn*, as a value from 1 (or 01) to 24. If your keyboard has only 12 PF keys, specify *nn* as a value from 1 (or 01) to 12.
2. SET PFnn TAB is not supported on a 3101 device. The 3101 has a tab function incorporated in its hardware.
3. SET PFnn TAB is an immediate (nondisplaying) command: its function is processed immediately upon pressing the appropriate program function key. If you insert the keyword DELAY after the PFnn operand, it is ignored; the program function is processed immediately.
4. To ensure the definition of a PF key for the TAB function, you must start with the keyword TAB, followed by at least one blank, followed by at least one tab setting that you want to associate with the PF key. The lack of a tab setting identifies the string TAB as PF key data.
5. If you press a PF key that has not been set, the following message is displayed:

```
PFnn UNDEFINED
```

### Messages

- HCP026E Operand missing or invalid

## SET PORT GROUP



### Authorization

Privilege Class: B

### Purpose

Use the SET PORT command to define or change the OSA-Express devices that make up a link aggregation group and to set the attributes of a link aggregation group. For more information about [Virtual Networking Options](#) in VM, see *z/VM: Connectivity*.

If the SET PORT GROUP command is issued with the JOIN, LACP or INTERVAL operand, and the group does not exist, it will be created and will remain until the next time z/VM is IPLed or the group name is deleted with a SET PORT groupname DELETE command. For options not specified, the following defaults will be used: LACP ACTIVE EXCLUSIVE INTERVAL 300.

### Operands

#### *groupname*

is the name of the link aggregation group to be affected. The *groupname* is a single token (1-8 alphanumeric characters) that identifies a link aggregation group. This groupname is associated with a virtual switch via the DEFINE VSWITCH command or statement in the System Configuration file. The association can be changed with the SET VSWITCH command.

#### **JOIN**

indicates that the devices (*nnn* or *nnnn.Pn*) specified will join the specified group. If the group does not exist, it will be created and will remain until the next time z/VM is IPLed or the group name is deleted with a SET PORT *groupname* DELETE command.

**nnn****nnnn.Pn**

is a real device number or a real device number and OSA-Express port number of a QDIO OSA-Express device to be affected within the specified group. Specify each real device number as a hexadecimal number. If specifying a real device and an OSA-Express port number, specify the real device number as a hexadecimal number between X'0001' and X'FFFD' followed by a period (.), the letter 'P' (or 'p') followed by the port number as a hexadecimal number between X'0' and X'F'. For example, to specify port 1 for RDEV 300, specify "300.P1". The value of the port number depends on how many ports the OSA-Express hardware adapter supports. If the port number is not specified, it will default to port 0.

You can specify a maximum of 8 real device numbers. If you specify more than one device number, each must be separated from the others by at least one blank. Each real device number represents a trio of devices. For example, specifying JOIN 111 means that devices 111-113 are used to provide the connection to the real hardware LAN segment.

When the port group has been defined with the SHARED option, some devices on the adapters containing the specified devices are reserved for system use. See usage note [3](#) for information about device number assignment for shared port groups.

See Usage Note [“1” on page 1612](#). for more information about the requirements for the OSA-Express devices.

**LEAve**

indicates that the devices (*rdev*) specified will leave the specified group.

**DELeTe**

indicates that the *groupname* definition will be removed from the system. You can not delete a group that is being used for an active virtual switch. Use SET VSWITCH DISCONNECT to make the virtual switch inactive and then SET VSWITCH NOGROUP to drop the association with the group.

A SET PORT GROUP DELETE for a port group configured with the SHARED option has global scope. A copy of the shared port group exists in each member of the IVL domain and ALL port groups with that name are deleted. You can not delete a shared port group that is in use on any other system.

**LACP ACTIVE****LACP INACTIVE**

LACP ACTIVE mode indicates that the virtual switch will initiate negotiations with the physical switch via the Link Aggregation Control Protocol (LACP) and will respond to LACP packets sent by the physical switch. LACP INACTIVE indicates that aggregation is to be performed, but without LACP. It is only useful when the physical switch is also operating without LACP. In this mode, there is no negotiation between the physical switch and the virtual switch.

Ensure the same LACP setting exists on both the virtual switch and the physical switch.

LACP ACTIVE is the default for a *groupname* until changed by a SET PORT GROUP LACP command.

You cannot change the LACP mode when the port group is associated with an active virtual switch. Use SET VSWITCH DISCONNECT to inactivate the virtual switch and then SET PORT GROUP to change the LACP mode followed by SET VSWITCH CONNECT.

SHARED may not be specified with LACP INACTIVE.

**EXClusive**

An exclusive port group is a set of OSA-E features configured so that each OSA-E feature supports only a single VSWITCH QDIO connection. In other words, no sharing of the OSA-Express adapter is permitted.

Once the port group is defined with the EXCLUSIVE option, it cannot be changed to a SHARED port group without deleting the port group.

**SHARed**

When LACP ACTIVE SHARED is specified, it defines a shared port group. A shared port group is a set of OSA-E features configured so that one or more global virtual switches and global virtual switch members share the OSA-Express adapter(s) that make up the port group. This defines a Multi-VSwitch LAG (Link Aggregation Group) configuration.

## SET PORT GROUP

A port group defined with the SHARED option can be used only with virtual switches defined with the GLOBAL option. The virtual switches can be in the same or different LPARs residing in the same CEC.

The port group has a global scope which means it provides connectivity across multiple virtual switches and multiple systems running z/VM. It is propagated to all other systems in the same IVL domain. In other words, the Control Program creates a copy of the shared port group with the same name on all other systems running z/VM that are in the domain (unless the shared port group member has already been defined with that name). A change to the shared port group on one system affects all copies of the shared port group on all other systems in the IVL domain.

A port group defined with the SHARED option provides a single point of control for all virtual switches using the shared port group. For example, a SET PORT GROUP groupname LEAVE command for a shared port group causes the specified port to be removed from **all** virtual switches on **all** IVL members using a copy of the port group with that name.

Because of the global scope, once the port group is defined with the SHARED option, it cannot be changed to an EXCLUSIVE port group without deleting the port group. In addition, the LACP mode may not be changed to INACTIVE.

See [Multi-VSwitch LAG Configuration](#) in *z/VM: Connectivity* for more information.

### **INTERVAL nnnn**

#### **INTERVAL OFF**

indicates the interval to be used by the Control Program (CP) when doing load balancing of conversations across multiple links in the group. It is specified as the number of seconds between load balancing operations across the link aggregation group. It must be a decimal number between 1 and 9990 or the keyword OFF. The interval value is rounded up to the next multiple of thirty seconds. INTERVAL OFF indicates that no load balancing is done by CP.

INTERVAL 300 is the default for a groupname until changed by a SET PORT GROUP INTERVAL command.

When the port group is configured to use the Independent Load Balance management method, the scope of the SET PORT GROUP INTERVAL setting is local to the system it is issued on.

When the port group is configured to use the Collaborative Load Balance management method, the scope of the SET PORT GROUP INTERVAL setting is global (affects all systems within the IVL).

### **LOADBALANCING INDEPENDENT**

Each VSwitch that uses a Link Aggregated Group (LAG) to provide connectivity to an external network is responsible for balancing its own bandwidth across all the OSA-Express adapters within the group. By default, exclusive port groups will use the Independent Load Balance management method.

When the port group is shared, the scope of the SET PORT GROUP LOADBALANCING setting is global (affects all systems within the IVL).

### **LOADBALANCING COLLABORATIVE**

The Collaborative Load Balance management method will consider and serialize its operation with all the sharing VSwitches. By default, a shared port group will use Collaborative Load Balance Management Method.

The Collaborative Load Balance management method can only be used when all active systems connected to the same IVL Domain have support for Collaborative Load Balance. If any z/VM System is added or already exists in the domain which doesn't have support for Collaborative Load Balance, then all Shared Port Groups within the domain will be automatically forced to use the Independent Load Balance management method.

Collaborative Load Balance is only allowed for shared port groups, and the scope of the SET PORT GROUP LOADBALANCING COLLABORATIVE setting is global (affects all systems with the IVL).

## **Usage Notes**

1. The following restrictions pertain to the OSA devices in the specified group:
  - The OSA port must support IEEE802.3ad Link Aggregation.

- To aggregate multiple OSA ports within a group, all ports in the group must connect to the same physical LAG group, run in full-duplex mode and operate at the same speed.
2. The following pertain to the OSA devices in an exclusive port group, that is, one that is configured with the EXCLUSIVE option:
    - When an OSA port is active in a link aggregated group, it can't have more than a single QDIO connection established on the adapter (CHPID). Any requests to establish a QDIO connection using another device number on the same CHPID in any logical partition will fail until the OSA port is no longer active within the group. Likewise, an OSA port with an active QDIO connection on any device number on the same CHPID will not be able to join a link aggregated group until there are no active QDIO connections on the port.
    - When the port group is configured as LACP Inactive, all ports in the group must support the network disablement function. Cards that do not support the network disablement function may be used only in a port group configured with LACP Active.
  3. The following pertain to the OSA devices in a shared port group, that is, one that is configured with the SHARED option:
    - Another shared port group with the same name defined anywhere in the IVL Domain must have a matching configuration. In fact, the Control Program propagates the shared port group to all other members of the domain if a copy of the port group does not already exist.
    - When a System Administrator specifies a device number on the SET PORT GROUP JOIN command three devices (subchannels) associated with the OSA Express adapter are reserved for system use. The Control Program chooses three devices from within the range of those associated with the OSA Express adapter, starting with the first (most significant) available device number.  
Devices reserved for system use are shown with "SYSTEM" status in the Query OSA display.
    - When a specific OSA Express adapter is configured in a shared port group, a set of three device numbers is reserved in each z/VM system that has an active IVL connection.
    - A shared port group may be shared by multiple global virtual switches that reside in the same system. Each shared port group is assigned an instance (0,1,2,3) when configured to a global virtual switch.
    - The administrator can use either port associated with a multiple-port OSA Express adapter. Once a port is activated in a shared port group, it is no longer available for guest usage -- the ATTACH command cannot be used to specify devices that are reserved for system use. Likewise, a network connection established by a guest on a specific port prevents its use in a shared port group.
    - When a shared port group is in use on a global virtual switch, an RDEV list may not be used for the virtual switch to provide back up connectivity. Instead, back up connectivity is provided by a different global virtual switch member using the shared port group.
    - A shared port group may be used by up to 64 global virtual switch members across the IVL domain. The members may be in the same or different global virtual switches.
  4. The maximum number of groups that can be defined to CP is 128.
  5. It is invalid to create a groupname of ALL.
  6. The number of devices identified by the RDEV keyword on the DEFINE or SET VSWITCH command combined with the number of devices in an associated GROUP may not exceed eight devices.
  7. When the virtual switch has been defined with the GROUP attribute, any devices identified by the RDEV keyword (on DEFINE or SET VSWITCH) are used for failover in the event of a real switch failure of the link aggregation group. Failover in this environment will be to a single OSA-Express device connected to a second real switch. The virtual switch will recognize when the port group is available again and will initiate recovery to the port group member(s). When the virtual switch has been defined with GROUP and RDEV NONE, it means that there is no link aggregation group failover in the event the real switch should fail.
  8. The port number specified by *nnnn .Pnn* will be used by the TCP/IP stack (controller) when initializing the device. If the port number is not specified, it will default to port 0. If the port number is not supported, initialization of the device will be terminated.

### Examples

To define a group named LINGROUP that contains multiple devices, specify the following:

```
set port group lingroup join B00 C00 D00
```

### Responses

Response 1:

```
Port group groupname is {created|updated|deleted}
```

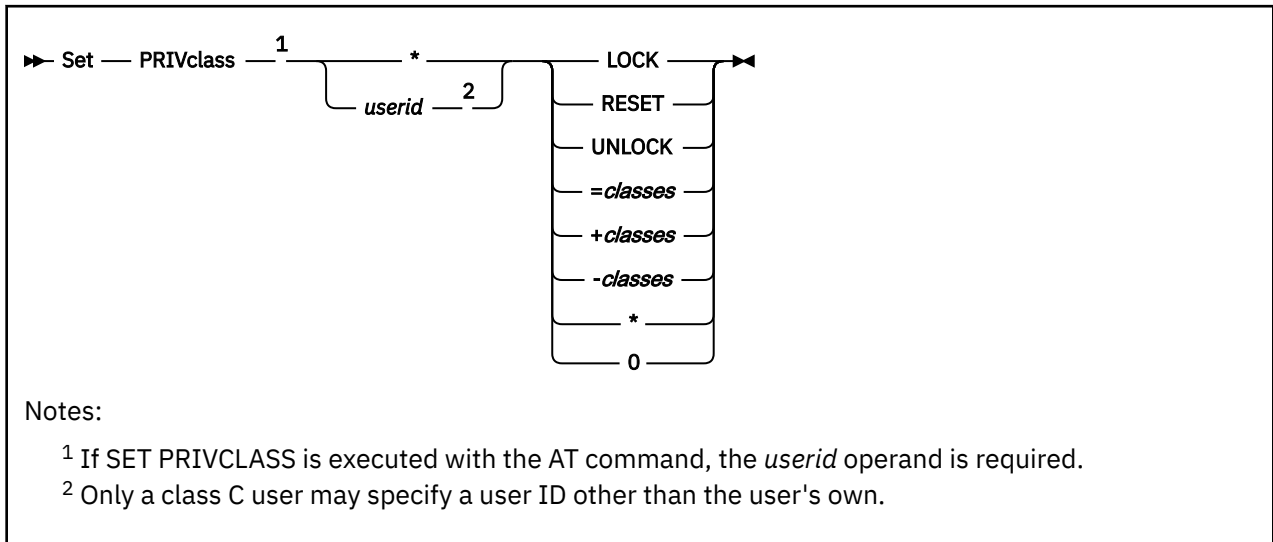
is the normal response from this command.

### Messages

- HCP002E Invalid operand - *operand*
- HCP2768E Missing *groupname*
- HCP2783E Port group *groupname* does not exist.
- HCP2799E SET PORT GROUP change is not allowed
- HCP2838I Device *device* specified for port group *groupname* is already attached to user *username*.
- HCP2838I Device *device* specified for VSWITCH *switchname* BRIDGEPORT is not a bridge-capable HiperSockets device.
- HCP2838I Port *portnumber* specified for device *device* is invalid; maximum port number is *maxportnumber*.
- HCP2857E SET PORT GROUP not performed; detach is in progress for associated VSWITCH SYSTEM *switchname*.
- HCP2857E SET PORT GROUP not performed; device *device* is already defined in port group.
- HCP2857E SET PORT GROUP not performed; device *device* not defined in port group.
- HCP2857E SET PORT GROUP not performed; group is defined to VSWITCH SYSTEM *switchname*.
- HCP2857E SET PORT GROUP not performed; group limit exceeded.
- HCP3001E Device *device* is defined for both port group *groupname* and VSWITCH SYSTEM *switchname*.
- HCP3162E SET PORT GROUP is not performed. No changes are permitted to LACP because the port group was defined with the SHARED option.
- HCP3162E SET PORT GROUP is not performed. Cannot change port group from EXCLUSIVE to SHARED.
- HCP3167E [ DEFINE VSWITCH *switchnm* | SET VSWITCH *switchnm* | SET PORT GROUP *groupname* ] failed. The IVL virtual switch does not exist or its UPLINK port is not connected.
- HCP3173E Device *device* is reserved for system use.
- HCP3178E [ VSWITCH SYSTEM *switchnm* | PORT GROUP *groupname* ] creation is deferred. The IVL virtual switch does not exist or its UPLINK port is not connected.
- HCP3180E SET PORT GROUP for shared port group *groupname* not performed; device *device* does not exist.
- HCP3183E SET PORT GROUP for shared port group *groupname* not performed; device *device* is not a real OSA-Express device.
- HCP3184E SET PORT GROUP for shared port group *groupname* not performed; all available devices on OSA PCHID *pchid* are already in use.
- HCP3186E [DEFINE VSWITCH GROUP | SET PORT GROUP | SET VSWITCH GROUP] for shared port group *groupname* not performed; devices on OSA PCHID *pchid* do not exist or are not available.
- HCP3194I Creating [ VSWITCH *switchnm* | PORT GROUP *groupname* ] now that the IVL Virtual switch is connected.

- HCP3197E SET PORT GROUP not performed; system *systemname* is busy executing a cross IVL Domain CP command.
- HCP3198E SET PORT GROUP not performed; an unexpected internal error occurred.
- HCP3200E SET PORT GROUP for shared port group *groupname* not performed; port groups cannot span CECs.
- HCP3220E SET PORT GROUP for shared port group *groupname* not performed; OSA PCHID *pchid* is used by another port group.
- HCP3223E SET PORT GROUP for port group *groupname* not performed; COLLABORATIVE is not allowed for an exclusive port group.
- HCP3225E IVL DOMAIN does not support Collaborative Load Balancing.
- HCP6704E Missing token at end of line
- HCP6706E Invalid device address - *rdev*
- HCP6706E Invalid device address or port number - {*vdev* | *portnumber*}
- HCP6706E Invalid groupname - *groupname*

## SET PRIVCLASS



### Authorization

Privilege Class: Any, C

### Purpose

Use SET PRIVCLASS to control future SET PRIVCLASS commands or to temporarily change the set of privilege classes for a logged-on user. If you are a class C user, you can set the privilege classes for any virtual machine. If you are not a class C user, you can set the privilege classes only for your own virtual machine.

When you log on, your user ID's privilege classes are set to the privilege classes granted to you in your user directory entry. If a class C user sets the privilege classes of another user ID or selects classes beyond those defined in the user directory, CP generates an accounting record.

To use the SET PRIVCLASS command, the FEATURES ENABLE SET\_PRIVCLASS statement must be specified in the system configuration file. For more information on the [FEATURES Statement](#), see *z/VM: CP Planning and Administration*.

### Operands

**\***

#### *userid*

identifies the user whose privilege classes are to be set. An asterisk (\*) specifies that the command applies to the command issuer's user ID.

#### **LOCK**

prevents any SET PRIVCLASS commands from being issued against the target until an UNLOCK is issued, if either the issuer has class C privileges or the target user ID is the issuer's.

#### **RESET**

returns the target user ID's privilege classes to those currently specified in the CP directory if the PRIVCLASS is not locked against SET PRIVCLASS processing and either the issuer has class C privileges or the target user ID is the issuer's.

#### **UNLOCK**

allows SET PRIVCLASS commands to be issued against the target if either the:

- Issuer is not the same as the target and has class C privileges



- Target user ID is the issuer's. If the target user ID is the issuer's ID, the issuer must confirm the issuer's intentions by responding with '1' for this command to complete successfully.

A SET PRIVCLASS \* UNLOCK command issued by a disconnected virtual machine is rejected because there is no terminal to enter a response.

#### **=classes**

sets the specified privilege class or classes for the target user ID, if the user ID is not locked against SET PRIVCLASS processing and either the issuer has class C privileges or the target user ID is the issuer's and the issuer is authorized in the directory for the requested classes.

#### **+classes**

adds the specified privilege class or classes for the target user ID, if the user ID is not locked against SET PRIVCLASS processing and either the issuer has class C privileges or the target user ID is the issuer's, and the issuer is authorized in the directory for the requested classes.

#### **-classes**

removes the specified privilege class or classes from the target user ID, if the user ID is not locked against SET PRIVCLASS processing and either the issuer has class C privileges or the target user ID is the issuer's.

#### **\***

indicates all classes, A-6.

#### **0**

indicates no classes.

## Usage Notes

1. Use the SET PRIVCLASS command to change your privilege classes to a subset of those defined in your directory entry. Users with class C privilege can exceed the privilege classes in the system directory. For example, users with class C privilege may enter a SET PRIVCLASS command to change another user ID's privilege classes.

## Responses

Response 1:

If you specify the LOCK option and the command completes successfully, you see something like this:

```
DAVID's privilege classes have been locked.
```

Response 2:

If you specify the UNLOCK option and the command completes successfully, you see something like this:

```
DAVID's privilege classes have been unlocked.
```

Response 3:

When you specify other options, CP displays the new privilege classes of the target user ID in its response. For example, user ID DAVID had privilege classes ABCEG, and you entered:

```
set priv david -ab
Privilege class(es) for user DAVID
  Currently: CEG
  Directory: ABCDEFG
The privilege classes are not locked against changes.
```

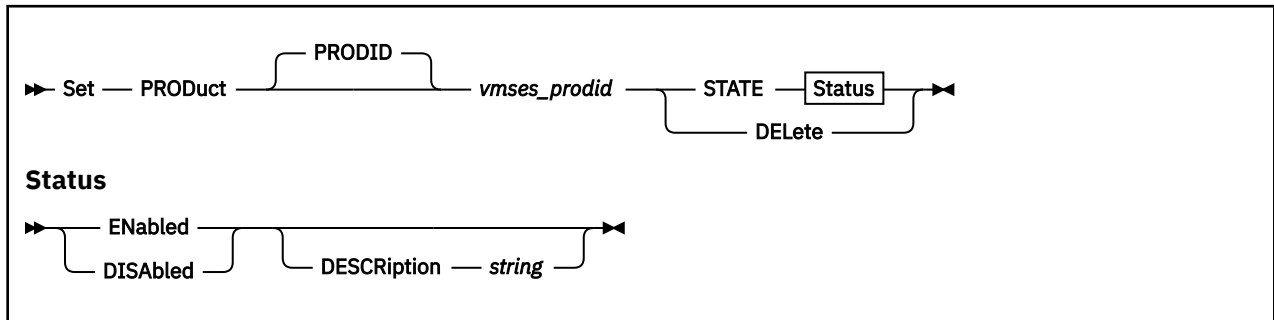
## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP003E Invalid option - *option*

## SET PRIVCLASS

- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP028E Class missing or invalid
- HCP045E *userid* not logged on
- HCP052E Error in CP directory
- HCP053E *userid* not in CP directory
- HCP1013E An invalid operand was supplied for CLASS - *classes*
- HCP1022A Enter 1 if you wish to have your privilege classes unlocked.
- HCP1023E An error was detected while processing the user's response. Please retry the command.
- HCP6750E *userid*'s privilege classes {are|are not} locked.
- HCP6754E Command combination may not be issued by a disconnected user

## SET PRODUCT



### Authorization

Privilege Class: C, E

### Purpose

Use SET PRODUCT to define a product or feature to the z/VM system and state whether the product can run on the system.

- Allows the user to dynamically add PRODUCT statements to the running system.
- Allows the user to change the state or description of an existing PRODUCT statement.
- Enables a user to DELETE and existing PRODUCT statement.

### Operands

#### PRODIG *vmstes\_prodid*

is the identifier used by VMSES/E to install and service the given product. *vmstes\_prodid* must be 7 or 8 characters.

#### STATE ENabled

identifies this product or feature as being licensed to execute on this system.

When a product is defined with STATE ENABLED, it indicates that the product is authorized by the installation to run on this system. It does not mean that the product is installed or is being used.

#### STATE DISAbled

identifies this product or feature as not being licensed to execute on this system. The product may or may not be physically installed on the system.

#### DESCRiption *string*

is a string of data that identifies additional information regarding the product.

*string* is a 1-255 character string (including any imbedded blanks and special characters associated with the *vmstes\_prodid* entered).

#### DELeTe

removes from the system any definition information for the product ID, *vmstes\_prodid*. Description information stored with the PRODUCT statement and the current STATE information is lost.

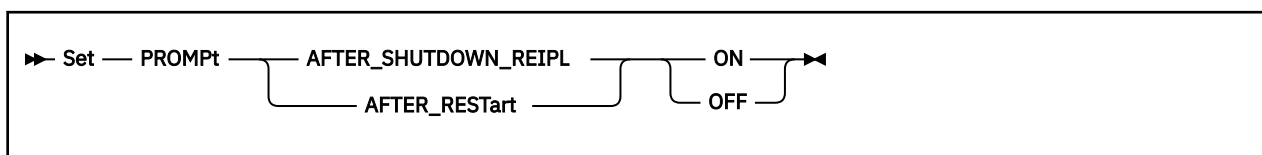
### Usage Notes

1. Each product is identified by a unique *vmstes\_prodid*. When you issue a SET PRODUCT command, it overwrites any previous definition, including the description.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP1001E An operand is missing for DESCRIPTION
- HCP1003E Product *prodid* was not found.
- HCP1639E The product definition for *prodid* may not be added, altered or deleted.
- HCP2768E Missing {product|product state}
- HCP6704E Missing token at end of line
- HCP6706E Invalid {string *string*|product *string*|product state *string*}

## SET PROMPT



### Authorization

Privilege Class: B

### Purpose

Use SET PROMPT to define whether CP will prompt the operator for startup information following a CP restart after an abend or a shutdown REIPL.

### Operands

#### AFTER\_SHUTDOWN\_REIPL ON/OFF

specifies whether CP will prompt the operator for startup information on a shutdown REIPL or not.

#### AFTER\_REStart ON/OFF

specifies whether CP will prompt the operator for startup information on a CP restart after an abend or not.

### Usage Notes

1. Use the QUERY PROMPT command to display the current settings of the prompt operands.
2. Use the FEATURES ENABLE/DISABLE PROMPT configuration file statement to initialize the prompt operands at IPL time.
3. If the SET PROMPT command was not used or the FEATURES ENABLE PROMPT statement was not specified in the configuration file, the operator will not be prompted for start up information after a system restart or a shutdown REIPL.

### Responses

Response 1:

To tell CP to prompt the operator on shutdown REIPL, enter the following:

```
set prompt after_shutdown_reipl on
```

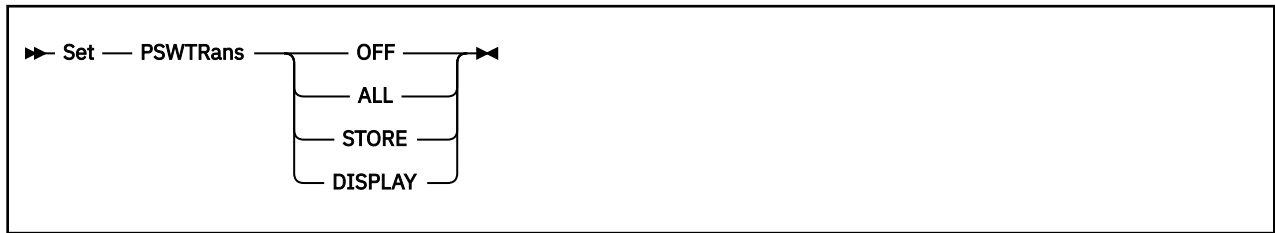
CP responds:

```
Command accepted
Ready;
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*

## SET PSWTRANS



### Authorization

Privilege Class: G

### Purpose

SET PSWTRANS allows you to issue DISPLAY, DUMP, and STORE CP Commands with the PSW option (instead of the PSWG option) and get results as if you had used the PSWG option. This is valid only on z/Architecture machines.

With PSWTRANS enabled, z/Architecture mode expands 64-bit PSW display and store data into 128-bit PSWs.

### Operands

#### OFF

indicates that the PSW option is not allowed for the DISPLAY, DUMP or STORE commands when the virtual machine is in z/Architecture mode.

#### ALL

indicates that the PSW option will be accepted for the DISPLAY, DUMP and STORE commands to reference or change PSW data in the 64-bit ESA/390 format when the virtual machine is in z/Architecture mode (64-bit PSW displayed and 64-bits accepted on STORE).

#### DISPLAY

indicates that the PSW option will be accepted for the DISPLAY and DUMP commands (but not STORE) to reference PSW data in the 64-bit ESA/390 format when the virtual machine is in z/Architecture mode.

#### STORE

indicates that the PSW option will be accepted for the STORE command (but not DISPLAY or DUMP) to expand PSW data in the 64-bit ESA/390 format when the virtual machine is in z/Architecture mode.

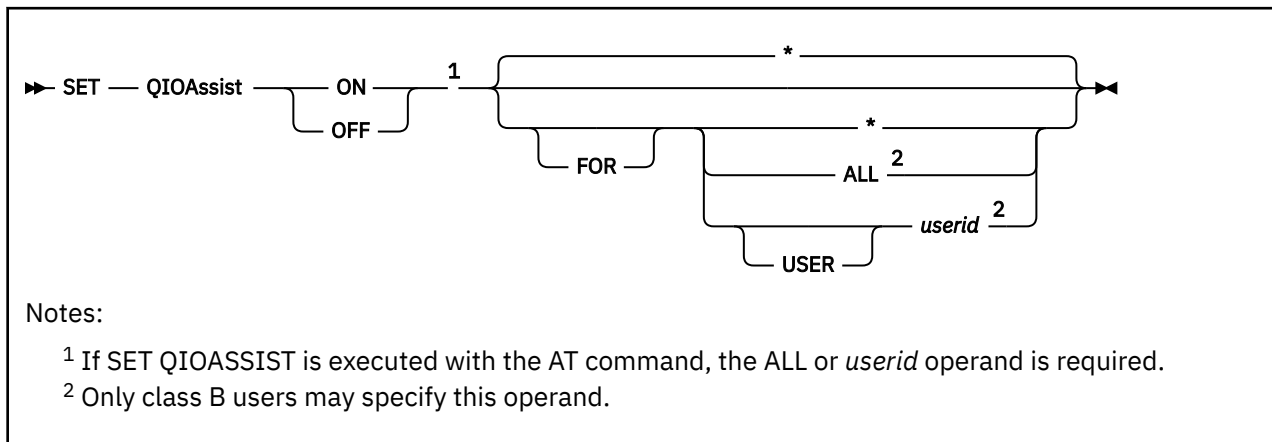
### Usage Notes

1. The PSWTRANS option is set to OFF when a user logs on and is not changed unless the user issues a SET PSWTRANS command.
2. To obtain the setting of the PSWTRANS option, use the QUERY PSWTRANS command.

### Messages

- HCP003E Invalid option - *option*
- HCP026E Operand missing or invalid

## SET QIOASSIST



### Authorization

Privilege Class: B, G

### Purpose

Use SET QIOASSIST to control the queue-I/O assist (QDIO performance assist for V=V guests) for a virtual machine. This interpretive-execution assist applies to devices that use the Queued Direct I/O (QDIO) architecture, HiperSockets devices, and FCP devices.

On processors that support the queue-I/O assist, there is a performance benefit for I/O performed with the queued-I/O protocols. With the aid of the assist, I/O interruptions can be passed directly from the hardware to the virtual machine. In addition, certain QDIO-related instructions can be interpretively executed by the processor, without z/VM's involvement.

To use queue-I/O assist, the following are required:

- The processor must support the queue-I/O assist. See *z/VM: General Information*.
- z/VM must be running in a logical partition and not as a guest.
- The assist must be globally enabled with the CP SET QIOASSIST ON ALL command.
- The virtual machine must be enabled with the CP SET QIOASSIST ON command.
- The device must not be attached to the virtual machine with the NOQIOASSIST option of the CP ATTACH command or the system DEDICATE directory statement.
- The virtual machine must not be using the CP TRACE command.

### Operands

#### ON

#### OFF

turns the queue-I/O assist on or off for a virtual machine.

#### ALL

enables or disables eligibility for all virtual machines to use the queue-I/O assist, independent of the individual settings for each user. The individual settings are not changed. When ALL is specified by a class B user, access to the queue-I/O assist is allowed or disallowed for all virtual machines, including any new users that might log on. When this global setting is OFF, it overrides any individual user's setting to allow the assist.

#### USER *userid*

specifies an individual user of the queue-I/O assist.

## SET QIOASSIST

\*

specifies the queue-I/O assist setting is for the command issuer's virtual machine.

### Usage Notes

- If QIOASSIST is set ON for all users, an individual user can set QIOASSIST OFF without changing the setting for other users.

**Note:** The reverse is not true; setting OFF for ALL overrides individual user settings.

- Setting QIOASSIST ON or OFF for ALL users may not immediately enable or disable the assist for all users. The command will take effect as soon as possible. Use the QUERY QIOASSIST to determine when the user's virtual machine is in the desired state.
- If z/VM is not running on a configuration that supports the queue-I/O assist, the SET QIOASSIST command is not supported and an error message is issued.

### Examples

The following command will disable userid test1 from using the queue-I/O assist.

```
set qioassist off test1
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP263E Too many operands were supplied for this command
- HCP1014E A required option is missing
- HCP2162I QIOAssist is not available



## SET QUICKDSP



### Authorization

Privilege Class: A

### Purpose

Use SET QUICKDSP to assign a user with immediate access to system resources.

**Note:** CP's virtual processor management has been improved so that no user stays in the eligible list more than an instant before being added to the dispatch list. Therefore the QUICKDSP option is now less meaningful, although it does get the virtual machine a different size elapsed time slice.

### Operands

#### *userid*

is the user ID of the virtual machine to receive the QUICKDSP designation.

#### **ON**

specifies that the user is to be added to the dispatch list immediately whenever the user has work to do, without waiting in the eligible list. Users not having this option may wait in the eligible list and be delayed for milliseconds or even seconds before running if there is a heavy load on the system. ON is the default.

#### **OFF**

specifies that the QUICKDSP option is to be taken away from this user. The user can subsequently encounter delays in the eligible list just as other users do.

### Usage Notes

1. The QUICKDSP designation is intended for selective use on virtual machines with critical response time requirements. The scheduler always moves a QUICKDSP user immediately into the dispatch list whenever it is ready to run, regardless of resource requirements and current system load. Indiscriminate use, therefore, increases response time overall and may severely affect maintenance of system storage.  
  
QUICKDSP is normally provided for use by selected service virtual machines interacting with several other users, thus having stringent response time requirements. RSCS and IUCV applications are common examples.
2. In some cases, the SET RESERVED command should be used with QUICKDSP, especially in storage-constrained environments. A user with the QUICKDSP attribute and reserved pages experiences little system overhead, further improving response time.
3. The QUICKDSP designation is given to users who are being forced or logged off from the system to expedite the logoff of those users.

### Responses

Response 1:

## SET QUICKDSP

```
USER userid: QUICKDSP={ON }  
                {OFF }
```

is returned to the command issuer.

### Messages

- HCP002E Invalid operand - *operand*
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on

# SET RDEVICE

---

## Authorization

Privilege Class: B

## Purpose

Use SET RDEVICE to add, change, or clear the system's definition of a set of real devices.

The following sections describe the syntax of SET RDEVICE commands for clearing real devices and for adding and changing these real devices:

- Advanced function printers
- Card punches
- Card readers
- Communication controllers
- DASDs
- Graphic display devices
- Impact printers
- Integrated communication adapters
- Special devices
- Tape units
- Unsupported devices
- 3270 displays
- 3800 printers.

## Usage Notes

1. Once you have defined a device at a certain address, you can replace that device with another device of a different class. The following are considered to be discrete device classes:
  - DASDs
  - Graphic display devices
  - Special devices (3088s, CTCAs, 37x5)
  - Special switching devices
  - Tape drives
  - Terminals
  - Unit record devices.
2. If you change the type of device for a given device address, the features previously defined for that device are returned to their defaults. When you change any feature other than device type for the device at a certain address, however, the other features remain the same. For example, assume you have an IBM 3800 Model 1 printer defined at address 0500 and CHARS GF10 and AFP YES specified and you want to replace it with an impact printer. When you use the SET RDEVICE TYPE IMPACT\_PRINTER command to tell CP the new printer is in place, CP resets all the other features to their defaults. If you leave the 3800 printer in place, however, and use SET RDEVICE to specify a flash overlay (SET RDEVICE ... FLASH *flashlib*), CP leaves all the other features unchanged.
3. You must use the VARY command to take a device offline before you can redefine it.

## SET RDEVICE

4. By default, CP issues a sense ID request to gather information about the device that you are defining or changing. If you prefer to explicitly define or change the device, you can stop CP from issuing the sense ID request by specifying the NOTSENSED operand. The only exception is the SET RDEVICE Unsupported Devices command. Because the device is unsupported, you must explicitly define that device. For information about the [DEVICES Statement](#), see *z/VM: CP Planning and Administration*.
5. The SET RDEVICE command cannot be used to change the device definition of any device that has a static device definition on an RDEVICE macro in the HCPRIO ASSEMBLE file.

### Responses

After CP has issued one of the following messages for each device address, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

### Messages

CP issues one of the following messages for each device address affected by a SET RDEVICE command. These two messages indicate that the SET RDEVICE command succeeded:

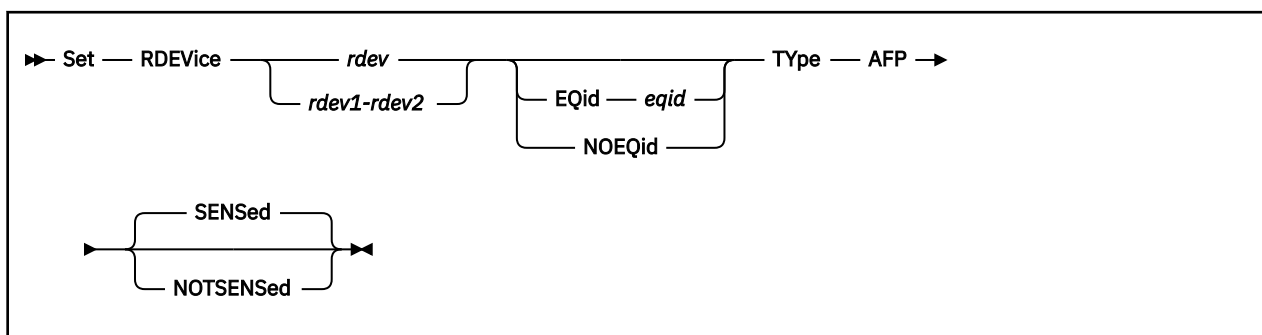
Messages:

- HCP6722I Characteristics of device *rdev* were set as requested.
- HCP6722I Created RDEV for device *rdev*.

The following messages indicate that the SET RDEVICE command failed:

- HCP1550E Device *rdev* device class does not match requested device class.
- HCP1550E Device *rdev* is not offline.
- HCP6580E Device *rdev* is active
- HCP6760E Device *rdev* is boxed.
- HCP6760E Device *rdev* is not offline.
- HCP6765E Device *rdev* cannot be changed because of an old HCPRIO definition.
- HCP6862E Device *rdev* cannot be changed or deleted because it is a base Parallel Access Volume for which an alias exists.
- HCP6870E Device *rdev* is attached to a guest
- HCP6895E RDEV *rdev* cannot be changed or deleted because it is static device definition from HCPRIO

## SET RDEVICE Advanced Function Printers



### Authorization

Privilege Class: B

### Purpose

Use this SET RDEVICE command to define advanced function printers at certain device addresses.

CP supports the following IBM advanced function printers using Advanced Function Printing licensed programs: 3825, 3828, and 3900

**Note:** Printers defined as advanced function printers that follow the CCU specifications can be used only as dedicated devices.

### Operands

#### *rdev*

#### *rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

#### **EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

#### **NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

#### **TYpe** **AFP**

specifies an advanced function printer (AFP) that follows the CCU specification. Printers defined as TYPE AFP can only be used as dedicated devices.

#### **SENSEd**

tells CP to use the information returned from a sense ID request to determine the characteristics of this printer. If omitted, the default is SENSED.

#### **NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the characteristics of this printer.

### Usage Notes

1. Printers defined as advanced function printers that follow the CCU specifications can be used only as dedicated devices.

## SET RDEVICE Advanced Function Printers

2. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
3. The system cannot generate EQIDs for devices that do not support RCD CCW.

### Examples

Example 1:

To specify an advanced function printer at device number X'0110', enter the following SET RDEVICE command:

```
set rdevice 0110 type afp
```

Example 2:

To specify three advanced function printers at device numbers X'0111', X'0112', and X'0113', enter this SET RDEVICE command:

```
set rdevice 0111-0113 type afp
```

Example 3:

To specify an advanced function printer at device number X'0110' with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0110 eqid ABCDEFGH type afp
```

Example 4:

To specify an advanced function printer at device number X'0110' and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0110 noeqid type afp
```

### Responses

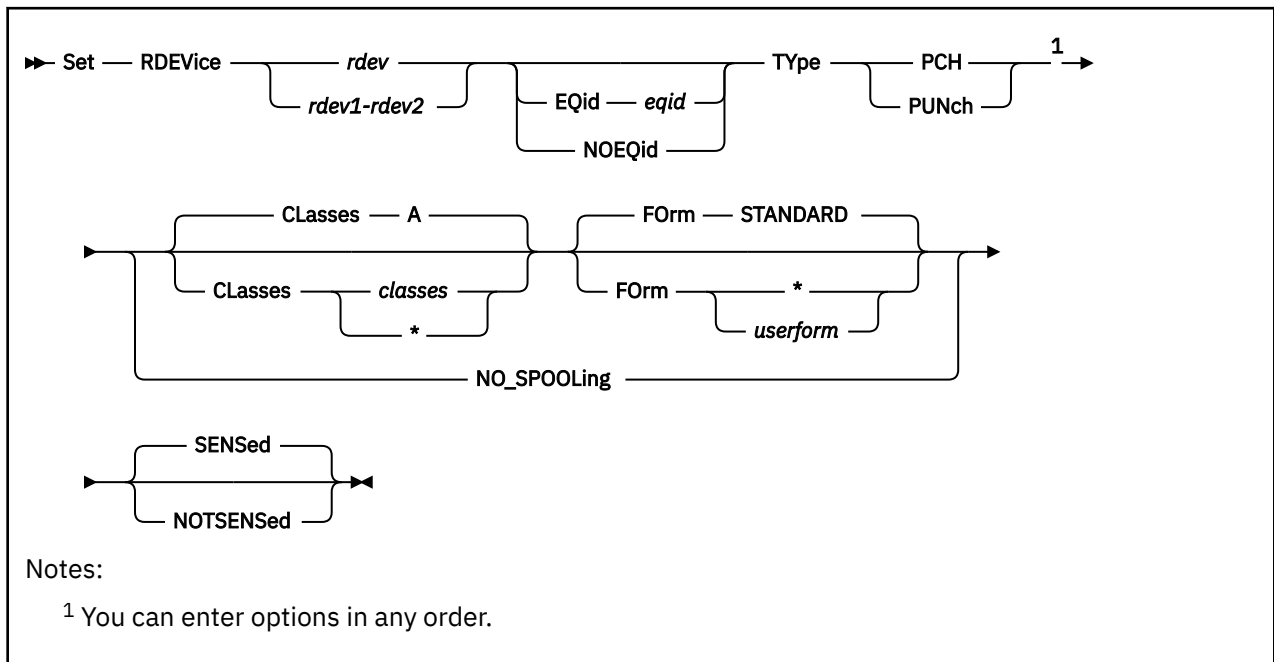
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE Card Punches



### Authorization

Privilege Class: B

### Purpose

Use this SET RDEVICE command to tell CP that card punches are at certain device addresses. CP supports the following card punches: 3525 and 3540

**Note:** The 3525 is supported only for punching; the printing features are not supported.

### Operands

#### *rdev*

#### *rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

#### **EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

#### **NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

#### **TYpe** **PUNch**

specifies the type of device. The 3525 is supported only for punching; the printing features are not supported.

#### **Classes** *classes*

specifies the output spooling classes. (To change these spooling classes without having to re-IPL, use the CP START command.)

## SET RDEVICE Card Punches

You can specify CLASSES in one of the following ways:

- Do not specify the CLASSES operand for unit record devices. Class A is the default.
- Enter CLASSES *c*[*c...*] to list up to 8 output spooling classes. The spooling class, *c*, is 1 alphanumeric character. You can specify multiple classes as CLASSES ABC and a single class as CLASSES A.
- Enter CLASSES \* to have the punch process files regardless of class.

### Form *userform*

is the current spooling form number that the punch can process. This form implicitly determines the default operator form for the real punch when the operator starts the device after a cold start without specifying a form. You can specify this operand in one of the following ways:

- Enter FORM *userform*, where *userform* is a 1- to 8-character user form number for the files the punch can process.
- Enter FORM \* to specify the punch can process files regardless of form number.
- Do not specify FORM. FORM STANDARD is the default.

### NO\_SPOOLing

tells CP not to use the unit record device for spooling.

### SENSEd

tells CP to use the information returned from a sense ID request to determine the characteristics of this card punch. If omitted, the default is SENSED.

### NOTSENSEd

tells CP not to use the information returned from a sense ID request to determine the characteristics of this card punch.

## Usage Notes

1. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCPO48E and the RDEV will not be created.
2. The system cannot generate EQIDs for devices that do not support RCD CCW.

## Examples

Example 1:

To specify a card punch at device number X'0110', enter the following SET RDEVICE command:

```
set rdevice 0110 type punch
```

Example 2:

To specify three card punches at device numbers X'0111', X'0112', and X'0113', and indicate they can process files regardless of form number, enter this SET RDEVICE command:

```
set rdevice 0111-0113 type punch form *
```

Example 3:

To specify a card punch at device number X'0110' with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0110 eqid ABCDEFGH type punch
```

Example 4:



To specify a card punch at device number X'0110' and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0110 noeqid type punch
```

## Responses

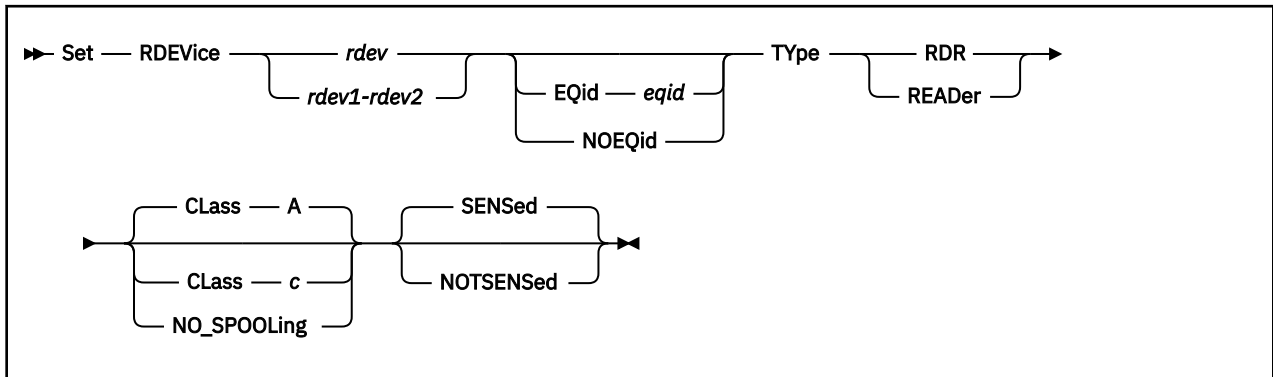
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP013E Conflicting option - *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE Card Readers



### Authorization

Privilege Class: B

### Purpose

Use this SET RDEVICE command to tell CP that real card readers are at certain device addresses.

### Operands

**rdev**

**rdev1-rdev2**

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

**EQid *eqid***

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

**NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

**TYpe RDR**

**TYpe READer**

specifies the type of device (card reader).

**Class *c***

specifies the spooling class that the card reader assigns to spool files it creates. (To change the spooling class the card reader assigns without having to re-IPL, use the CP START command.)

You can specify CLASS in one of the following ways:

- If you omit the CLASS operand for unit record devices, class A is the default.
- Enter CLASS *c*

The spooling class, *c*, is one alphanumeric character.

**NO\_SPOOLing**

tells CP not to use the unit record device for spooling.

**SENSEd**

tells CP to use the information returned from a sense ID request to determine the characteristics of this card reader. If omitted, the default is SENSED.

**NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the characteristics of this card reader.

**Usage Notes**

1. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
2. The system cannot generate EQIDs for devices that do not support RCD CCW.

**Examples**

Example 1:

To specify a card reader at device number X'0110', enter the following SET RDEVICE command:

```
set rdevice 0110 type reader
```

Example 2:

To specify three card readers at device numbers X'0111', X'0112', and X'0113' and tell CP not to use them for spooling, enter this SET RDEVICE command:

```
set rdevice 0111-0113 type reader no_spooling
```

Example 3:

To specify a card reader at device number X'0110' with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0110 eqid ABCDEFGH type reader
```

Example 4:

To specify a card reader at device number X'0110' and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0110 noeqid type reader
```

**Responses**

After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE CLEAR



### Authorization

Privilege Class: B

### Purpose

Use SET RDEVICE CLEAR to clear the information in the real device control block (RDEV) for a specific device or range of devices. After you issue this command, CP clears out all of the information in the RDEV except for the:

- Count of granted lock requests
- Count of deferred lock requests
- Device number
- Enqueued task
- Free storage RDEV indicator
- Missing Interrupt Handler control block
- Monitor flags
- RDEV lock
- Size of RDEV
- Subchannel disabled indicator
- Subchannel identifier
- Subchannel measurement block address
- Subchannel measurement block index
- Task requesting lock
- Valid device indicator

### Operands

*rdev*

*rdev1-rdev2*

specifies the real device number or a range of real device numbers that you want cleared. The variable *rdev* must be a hexadecimal number between X'0000' and X'FFFF'.

### Usage Notes

1. Only devices whose characteristics can be determined dynamically (sensed), can be directly varied online after a clear has been performed.
2. For those devices whose characteristics cannot be determined dynamically, another SET RDEVICE must be performed to define the device before it can be varied online.
3. Although the EQID and NOEQID operands are allowed, they are ignored when you specify the SET RDEVICE CLEAR command.

**Note:** To determine if the given device(s) can be dynamically sensed, see [The System Configuration File in z/VM: CP Planning and Administration](#).

## Examples

Example 1:

To clear the information in the RDEV for device 600 enter the following SET RDEVICE CLEAR command:

```
set rdevice 600 clear
```

Example 2:

To clear the information in a range of RDEVs starting at device 700 and continuing until device 710 enter the following SET RDEVICE command:

```
set rdevice 700-710 clear
```

## Responses

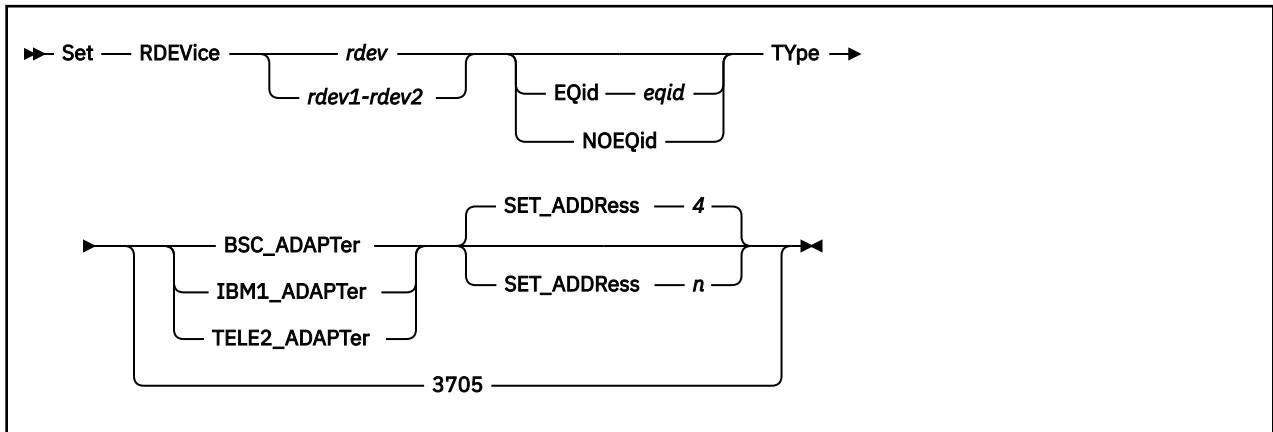
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE Communication Controllers



### Authorization

Privilege Class: B

### Purpose

Use this SET RDEVICE command to tell CP that line adapters or communication controllers are at specific device addresses.

### Operands

**rdev**

**rdev1-rdev2**

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

**EQid eqid**

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

**NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

**Type**

specifies the type of device.

**BSC\_ADAPTer**

specifies an IBM Binary Synchronous Terminal Control Type II for a 3745.

The BSCA adapter is supported as dedicated-only.

**IBM1\_ADAPTer**

specifies that an IBM Terminal Adapter Type I attaches a 1050 or 2741 to a 3745.

**TELE2\_ADAPTer**

specifies that a CPT-TWX (Models 33/35) Terminal, 3101, 3151, 3161, 3162, or 3163, attaches to a Line Interface Base Type I in a 3745.

**SET\_ADDRESS n**

is the set address (SAD) command to be entered for a telecommunication line attached to a control unit. The default is 4.

The values for *n* and the corresponding commands are:

- 0** SADZERO
- 1** SADONE
- 2** SADTWO
- 3** SADTHREE
- 4** (No SAD command is entered)

### 3705

defines the base address that loads the 3745 communications controller.

## Usage Notes

1. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
2. The system cannot generate EQIDs for devices that do not support RCD CCW.

## Examples

Example 1:

To specify an IBM Terminal Adapter Type 1 at device number 01F0, enter the following SET RDEVICE command:

```
set rdevice 01F0 type ibm1_adapter
```

Example 2:

To specify three Binary Synchronous Terminal Control Type II adapters at device numbers 01F1, 01F2, and 01F3, enter this SET RDEVICE command:

```
set rdevice 01F1-01F3 type bsc_adapter
```

Example 3:

To specify Binary Synchronous Terminal Control Type II adapters at device number 01F1 with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 01F1 eqid ABCDEFGH type bsc_adapter
```

Example 4:

To specify Binary Synchronous Terminal Control Type II adapters at device number 01F1 and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 01F1 noeqid type bsc_adapter
```

## Responses

After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

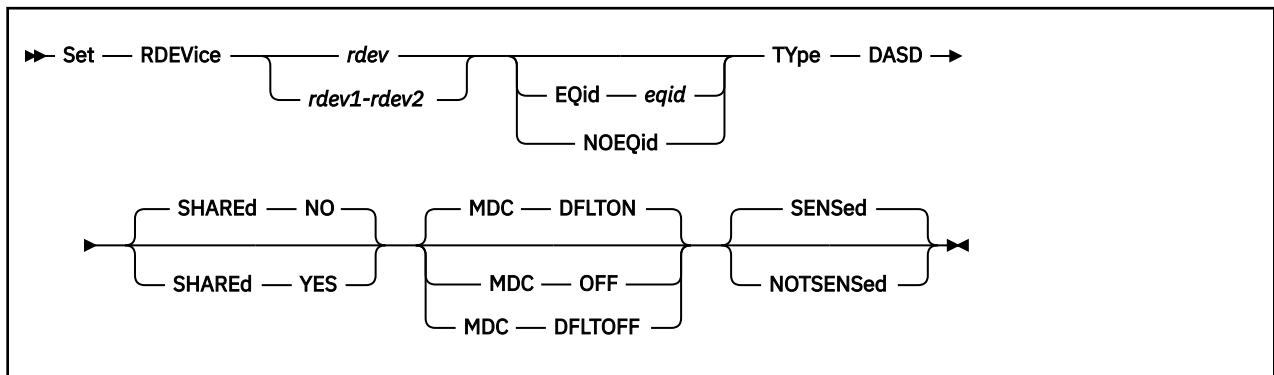
```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*



## SET RDEVICE DASD



### Authorization

Privilege Class: B

### Purpose

You can enter SET RDEVICE commands for the following DASD types: 3380, 3390 and 9336.

These DASD types respond to the sense ID request instruction. You only need to enter SET RDEVICE commands for them if you want to specify one of the following operands:

- SHARED YES
- MDC DFLTOFF
- MDC OFF
- NOTSENSED.

### Operands

#### *rdev*

#### *rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. A real DASD in the active configuration can be entered as a 4-digit hexadecimal device number between X'0000' and X'FFFF'. For a real DASD in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is defined. A range of devices cannot span subchannel sets. The maximum number of devices allowed within a range is 256.

#### **EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

#### **NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

#### **TYpe** **DASD**

tells CP the specified device or devices are DASDs.

#### **SHAREd**

tells CP to share the devices concurrently among multiple real systems.

In an SSI cluster, this setting specifies whether the devices are to be shared outside the SSI cluster.

## SET RDEVICE DASD

### **NO**

tells CP not to share the devices among multiple real systems. The default is NO.

### **YES**

tells CP to share the devices among multiple real systems.

### **MDC**

tells CP whether you want the device or devices to be cached in the minidisk cache.

### **DFLTON**

(the default if the SHARED option is not specified or the SHARED NO option is specified.) Tells CP to cache data on the real device in the minidisk cache except for minidisks for whom caching is specifically disabled via the MINIOPT NOMDC directory statement or the SET MDCACHE MDISK OFF command.

### **OFF**

(the default if the SHARED YES option is specified.) Tells CP to not cache data on the real device in the minidisk cache.

### **DFLTOFF**

tells CP to not cache data on the real device in the minidisk cache except for minidisks for whom caching is specifically enabled via the MINIOPT MDC directory statement or the SET MDCACHE MDISK ON command.

### **SENSEd**

tells CP to use the information returned from a sense ID request to determine the characteristics of this DASD. If omitted, the default is SENSED.

### **NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the characteristics of this DASD.

## Usage Notes

1. Not all DASD types are eligible for caching in the minidisk cache. If you specify MDC DFLTON or MDC DFLTOFF for a device that is not cache eligible, data on that device will not be cached.
2. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
3. The system cannot generate EQIDs for devices that do not support RCD CCW.
4. Specifying a 5-digit hexadecimal device number when defining an RDEV might place in into the active configuration. This would occur if there is not already a matching device number that is in the active configuration.

## Examples

### Example 1:

To specify a shared DASD with minidisk caching off at device number 0410, enter the following SET RDEVICE command:

```
set rdevice 0410 type dasd shared yes
```

### Example 2:

To define a non-shared DASD at device number 500 with minidisk caching enabled on a device only for those minidisks that have specifically enabled caching, use the following SET RDEVICE command:

```
set rdevice 500 type dasd mdc dfltoff
```

### Example 3:

To specify a shared DASD with minidisk caching off at device number 0252 with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0252 eqid ABCDEFGH type dasd shared yes
```

Example 4:

To specify a shared DASD with minidisk caching off at device number 0252 and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0252 noeqid type dasd shared yes
```

## Responses

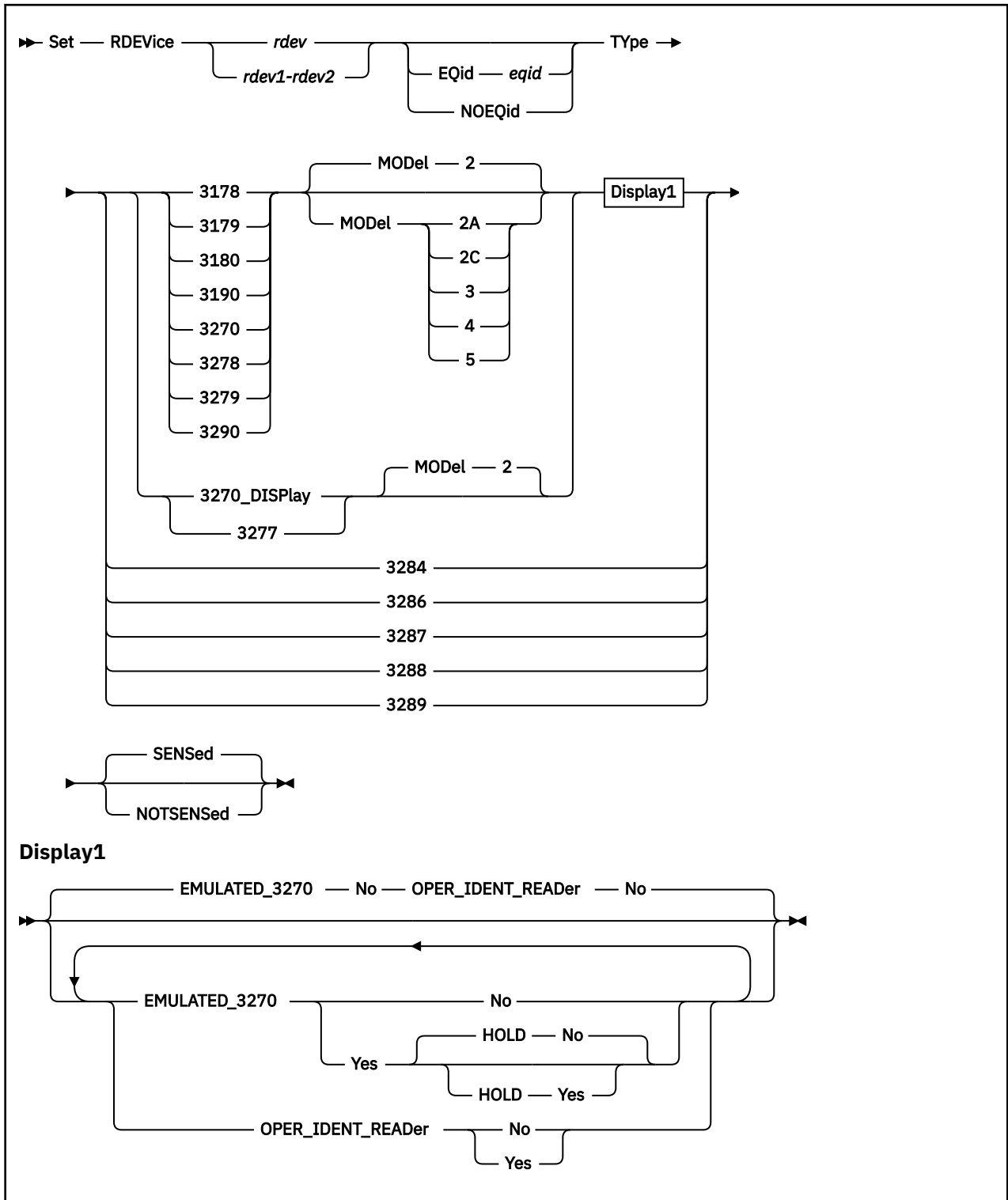
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE Graphic Display Devices



### Authorization

Privilege Class: B

## Purpose

Use this SET RDEVICE command to tell CP that IBM graphic display devices are at certain device addresses.

## Operands

*rdev*

*rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

**EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

**NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

**TYpe 3178, 3179, 3180, 3190, 3270, 3278, 3279, or 3290**

tells CP that the real device (or devices) that you are defining are graphic display devices. CP can determine dynamically all of these graphic display devices during initialization or when you vary on the device. You only need to specify these graphic display devices if you want to specify the extended attributes.

**MODeL 2, 2A, 2C, 3, 4, or 5**

is the model number of the graphic display device. If omitted, the default is Model 2.

**TYpe 3270\_DISPLAY or 3277**

tells CP that the real device (or devices) that you are defining are graphic display devices. CP can determine dynamically all of the 3270 graphic display devices during initialization or when you vary on the device. You only need to specify the 3270 graphic display devices if you want to specify the extended attributes.

**Note:** The 3270-PC is supported in control-unit terminal mode and distributed-function terminal (DTF) mode when defined as a 3270.

You must specify the 3277 display because its device type cannot be determined.

**MODeL 2**

is the model number of the graphic display device. If omitted, the default is Model 2.

**TYpe 3284, 3286, 3287, 3288, or 3289**

tells CP that the real device (or devices) that you are defining are graphic printer devices. CP can determine dynamically all of the graphic printer devices during initialization or when you vary on the device; therefore, you do not need to specify them.

**Note:** CP cannot sense these devices when they are attached to either a 3174 or a 3274 control unit. If these devices are attached to a 3174 or a 3274 control unit, each device must be defined in the system configuration file using the RDEVICE statement.

**EMULATED\_3270 Yes**

**EMULATED\_3270 No**

specifies whether the device is a TTY ASCII display terminal connected to the system through a 7171 ASCII DACU or ASCII Subsystem (an emulated 3270). NO is the default.

**Note:** The 3101 must be connected to a 7171 Control Unit. You must also specify EMULATED\_3270 YES.

**HOLD Yes****HOLD No**

specifies whether the display terminal telecommunications connection to the 7171 ASCII DACU or ASCII Subsystem is to remain after logging off or disconnecting. NO is the default.

**OPER\_IDENT\_READER Yes****OPER\_IDENT\_READER No**

specifies whether there is an operator identification card reader on a 327x display. If you specify OPER\_IDENT\_READER YES, the virtual machine operator can gain access to the system (log on) only by inserting a magnetically encoded card. You may ask each user ID to use a badge reader, but it is not required. You may use a badge reader as an additional security measure after a correct password has been entered, but it cannot replace the need to enter a correct password. If you do not want to have a card reader authorize access, do not enter OPER\_IDENT\_READER YES, or enter OPER\_IDENT\_READER NO. NO is the default.

**SENSEd**

tells CP to use the information returned from a sense ID request to determine the characteristics of this graphic display device. If omitted, the default is SENSED.

**NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the characteristics of this graphic display device.

### Usage Notes

1. *z/VM: General Information* lists all displays supported for this release.
2. Make sure the device numbers specified on the command correspond to the device numbers required for the category of display terminal in the controller.
3. Refer to *7171 Device Attachment Control Unit Reference Manual and Programming Guide* to determine which type of device (TYPE) to code for a TTY ASCII terminal attached to a 7171 DACU or ASCII Subsystem.
4. The 3178, 3179, 3180, 3190, 3278, 3279, 3290, 3284, 3486, 3287, 3288 and 3289 are defined primarily for use with second level systems. Second level, you could define a printer or display to be at a particular device number and then you could do a first level define of the device number. If the device is defined in the configuration file, the second level system would bring that device online immediately rather than the user having to define a device number first level, enter the SET RDEVICE command second level, and vary on the device second level.
5. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
6. The system cannot generate EQIDs for devices that do not support RCD CCW.

### Examples

Example 1:

To specify a 3277 display at device number 0210, enter the following SET RDEVICE command:

```
set rdevice 0210 type 3277
```

Example 2:

To specify three 3270 displays at device numbers 0211, 0212, and 0213, and not allow users to log on to them without an operator identification card, enter this SET RDEVICE command:

```
set rdevice 0211-0213 type 3270_display oper_ident_reader yes
```

Example 3:

To define:

- 48 3270 displays at device numbers 800 through 82F
- 16 3277 displays at device numbers 830 through 83F
- 31 3278 displays at device numbers 970 through 98E
- One 3287 display at device number 98F
- One 3290 display at device number 1010
- 15 3190 displays at device numbers 1011 through 101F
- 61 3278 displays at device numbers 200 through 23C, indicate the display is a TTY ASCII display terminal connected to the system through a 7171 ASCII DACU or ASCII Subsystem (an emulated 3270), and indicate the connection between this display and the 7171 is to remain after the user logs off or disconnects

use the following SET RDEVICE commands:

```
set rdevice 800-82f type 3270_display
set rdevice 830-83f type 3277
set rdevice 970-98e type 3278
set rdevice 98f type 3287
set rdevice 1010 type 3290
set rdevice 1011-101f type 3190
set rdevice 200-23c type 3278 emulated_3270 yes hold yes
```

Example 4:

To specify a 3277 display at device number 0252 with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0252 eqid ABCDEFGH type 3277
```

Example 5:

To specify a 3277 display at device number 0252 and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0252 noeqid type 3277
```

## Responses

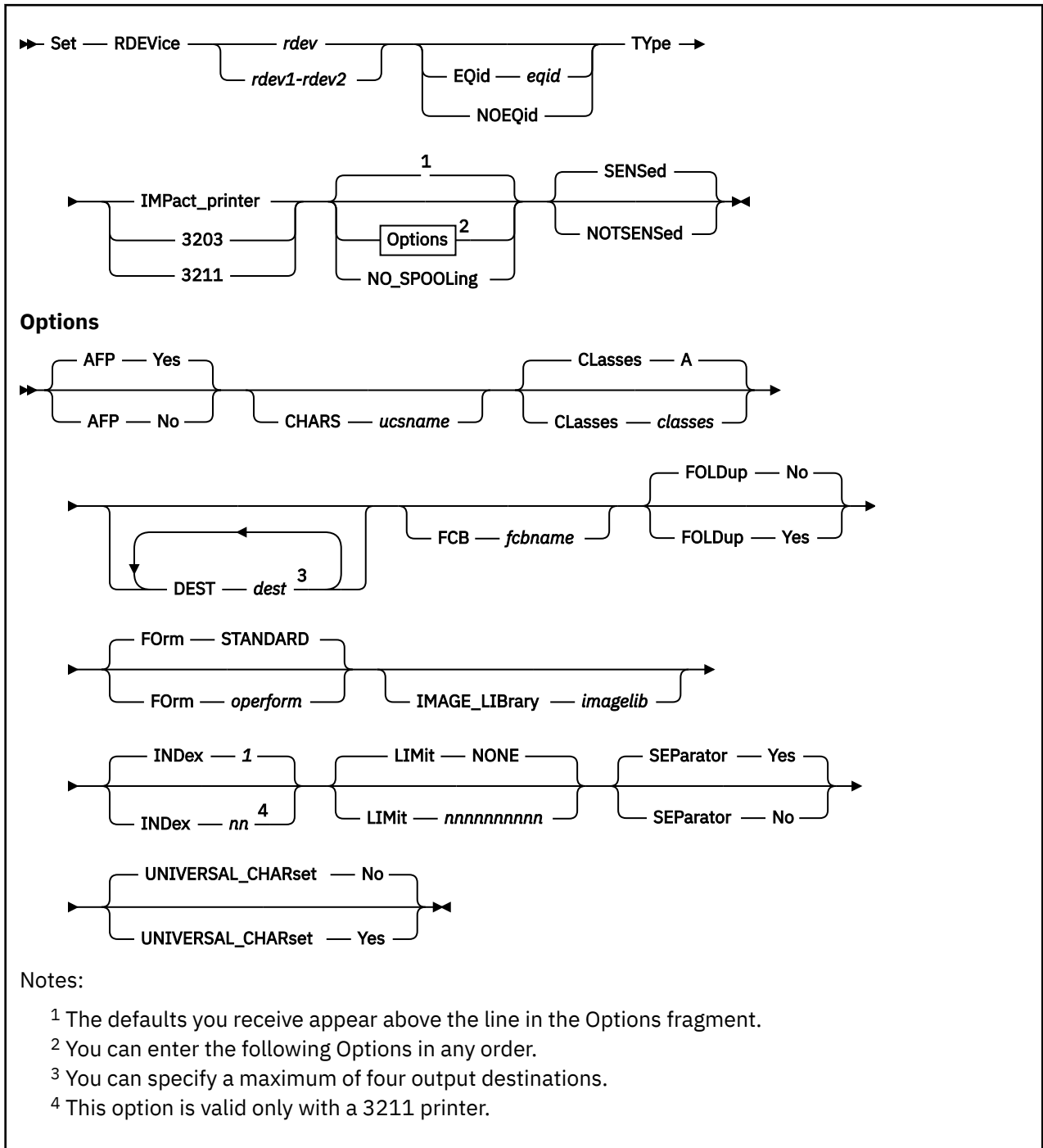
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP013E Conflicting option - *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE Impact Printers



### Authorization

Privilege Class: B

### Purpose

Use this SET RDEVICE command to define or change the following CP system-managed IBM impact printers: 3203, 3211, 3262, 4245, 4248, and 6262



**Note:** Only the 3203 Model 5 is supported. 3211 printers are only supported as emulated by other printers.

## Operands

### *rdev*

#### *rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

#### **EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

#### **NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

#### **TYpe IMPact\_printer**

tells CP you are defining one of the impact printers listed above.

#### **NO\_SPOOLing**

tells CP not to use the unit record device for printing.

#### **SENSEd**

tells CP to use the information returned from a sense ID request to determine the characteristics of this printer. If omitted, the default is SENSED.

#### **NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the characteristics of this printer.

#### **AFP Yes**

#### **AFP No**

specifies whether the printer is to process files with advanced function printer (AFP) characteristics, extended attribute buffers (XABs), or X'5A' channel command words (CCWs). The default is YES.

#### **CHARS** *ucsname*

For a printer with the UNIVERSAL\_CHARSET feature, CHARS *ucsname* specifies the 1- to 4-character suffix of the name of the default universal character set (UCS) buffer image. *ucsname* must correspond to one of the UCS images stored in the image library. For example, if you enter:

```
set rdevice 0003 type 3203 chars an
```

The image library must contain a member named 3203AN.

**Note:** If CP determines the printer is a 3262, the default for CHARS is P48. The default for 3203 is PCAN. The default for 3211 is A11.

#### **Classes**

specifies the output spooling classes the printer can print. (To change the spooling classes without having to take the printer off line, use the CP START command.)

#### **A**

is the default class.

#### **classes**

are the output spooling classes the printer can print.

You can specify CLASSES in one of the following ways:

- If you omit the CLASSES operand for unit record devices, class A is the default.

## SET RDEVICE Impact Printers

- Enter `CLASSES c[c...]` to list up to eight output spooling classes. The spooling class, *c*, is one alphanumeric character. You can specify multiple classes as `CLASSES ABC` and a single class as `CLASSES A`.
- Enter `CLASSES *` to have the printer process files regardless of class.

### **DEST** *dest*

specifies the output destination values the printer can print. (To change the spooling destination values without having to take the printer off line, use the CP START command.)

You can specify DEST in one of the following ways:

- If you omit the DEST operand, DEST OFF is the default.
- You can specify as many as four destination values by entering four different DEST operands, for example:

```
DEST PRINTER1 DEST PRINTER2 DEST PRINTER3 DEST PRINTER4
```

- Enter `DEST *` to have the printer process files regardless of destination identifiers.

### **FCB** *fcname*

specifies the 1- to 4-character suffix of the name of the default forms control buffer (FCB) to use after a cold start or force start. This must correspond to one of the FCB images added to an image library. For example, if you enter:

```
set rdevice 0003 type 3203 fcb fcb8
```

The image library must contain a member named 3203FCB8.

The number of lines to be printed on the separator page is determined by the FCB loaded on the printer. The default separator page contains 58 lines of data. If the page length defined by the FCB is less than the default separator page length, the separator page data must be customized in order to fit on a single page. This may be done using CP Exit points that are provided in separator page processing. See [z/VM: CP Exit Customization](#) for details.

### **FOLDup** Yes

### **FOLDup** No

specifies whether lowercase should be folded (translated) into uppercase. The default is NO.

### **Form** *operform*

is the current spooling form number that the printer can process. This form is the default operator form for the real printer. Specify this parameter when you are defining a new printer or when you are redefining an existing non-impact printer to be an impact printer. You can specify this parameter in one of the following ways:

- `FORM operform`

*operform* specifies a 1- to 8-character operator form number for the files the printer can process.

- `FORM *`

The asterisk indicates the printer can process files regardless of form number.

- `FORM STANDARD`

STANDARD indicates the type of paper to which the installation has assigned the name STANDARD. Each installation establishes its own set of form names, assigns those form names to types of paper, and tells the operations staff and end users the form names for the different types of paper.

FORM STANDARD is the default.

### **IMAGE\_LIBRARY** *imagelib*

specifies the image library to use after a cold start. The default for IMAGE\_LIBRARY is `IMAGnnnn`. *nnnn* is the device type number.

### **INDEX** *nn*

specifies the position at which to start printing. *nn* is a number from 1 to 31. The default is 1. If the defined printer is not a 3211, the INDEX option is ignored.

**LIMit nnnnnnnnnn****LIMit NONE**

specifies the maximum number of records of a file to be printed. You can specify up to 10 digits. NONE is the default.

**SEParator Yes****SEParator No**

specifies whether a separator is desired for output files. The default is YES.

**UNIVERSAL\_CHARset Yes****UNIVERSAL\_CHARset No**

tells CP the printer is a universal character set printer. The default is NO.

**Usage Notes**

1. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
2. The system cannot generate EQIDs for devices that do not support RCD CCW.

**Examples**

## Example 1:

To specify a 3203 printer at device number X'0110' to print only class A spool files (the default value), enter the following SET RDEVICE command:

```
set rdevice 0110 type impact_printer
```

## Example 2:

To specify a dedicated 3262 printer at device number X'0110', enter this SET RDEVICE command:

```
set rdevice 0110 type impact_printer no_spooling
```

## Example 3:

To specify a 3203 printer at device number X'0110' with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0110 eqid ABCDEFGH type impact_printer
```

## Example 4:

To specify a 3203 printer at device number X'0110' and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0110 noeqid type impact_printer
```

**Responses**

After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

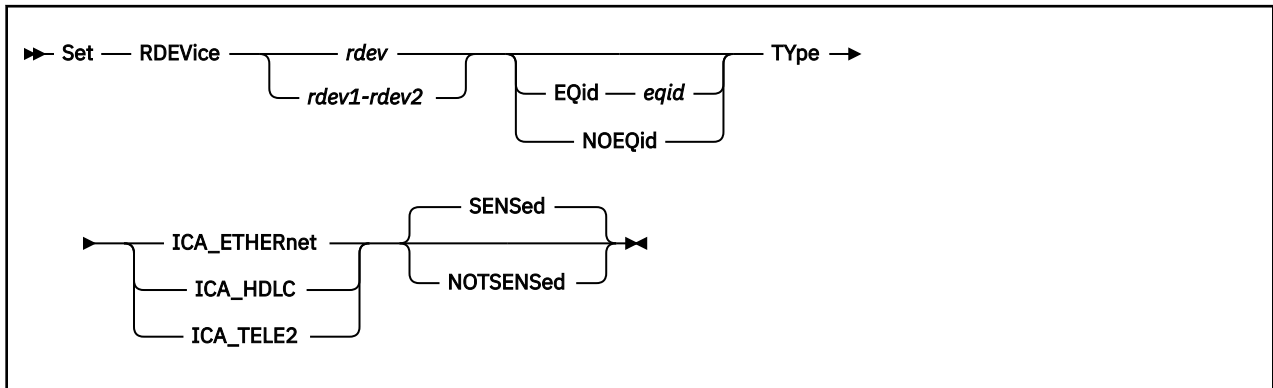
**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*

## SET RDEVICE Impact Printers

- HCP013E Conflicting option - *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*
- HCP6709E Too many items specified - *token*

## SET RDEVICE Integrated Communication Adapters



### Authorization

Privilege Class: B

### Purpose

Use the SET RDEVICE command to tell CP that integrated communication adapters are at certain device addresses.

### Operands

#### *rdev*

#### *rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

#### **EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

#### **NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

#### **TType** *ICA\_ETHERnet*

tells CP that the real device (or devices) that you are defining is a teleprocessing adapter for an 802.3 local area network.

#### **TType** *ICA\_HDLC*

tells CP that the real device (or devices) that you are defining is a teleprocessing adapter for a high level data link control.

#### **SENSEd**

tells CP to use the information returned from a sense ID request to determine the characteristics of this adapter. If omitted, the default is SENSED.

#### **NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the characteristics of this adapter.

### Usage Notes

1. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
2. The system cannot generate EQIDs for devices that do not support RCD CCW.

### Examples

Example 1:

To define 4 integrated communications adapters at device numbers 300 through 303 to an 802.3 Local Area Network, enter the following SET RDEVICE command:

```
set rdevice 300-303 type ica_ethernet
```

Example 2:

To specify an integrated communications adapter at device number 0252 with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0252 eqid ABCDEFGH type ica_ethernet
```

Example 3:

To specify an integrated communications adapter at device number 0252 and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0252 noeqid type ica_ethernet
```

### Responses

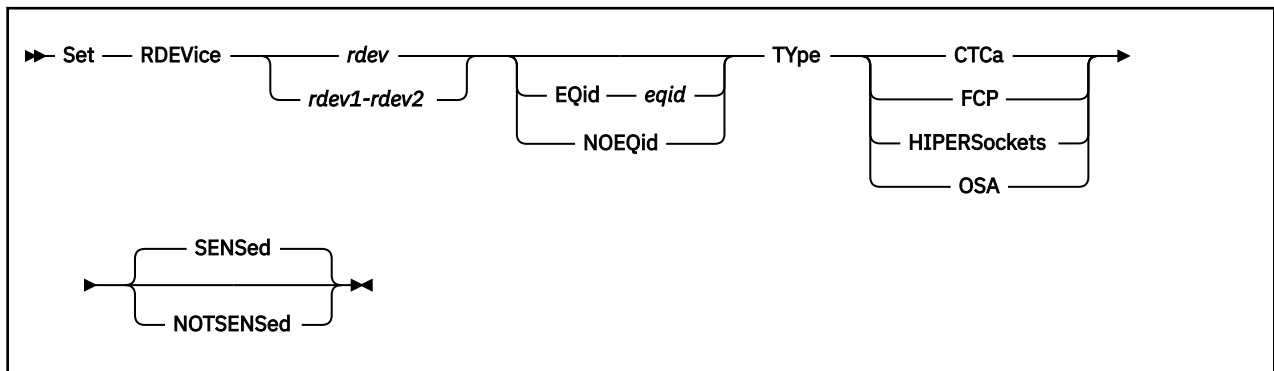
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE Special Devices



### Authorization

Privilege Class: B

### Purpose

Use this SET RDEVICE command to tell CP the device addresses of certain types of devices.

### Operands

*rdev*

*rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

**EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for CTCA, FCP, HiperSocket, and OSA devices, this EQID must be unique or be shared only by other devices of the same type.

**NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

**TYpe**

specifies the type of device.

**CTCa**

is a Channel-to-Channel device.

**Note:** Specify the 3737 Remote Channel-to-Channel Unit Model 2 as a CTCA.

**FCP**

is a Fiber-Channel-Protocol device.

**HIPERSockets**

is a HiperSockets device. To obtain status for a HiperSockets device, issue a Query OSA command.

**OSA**

is an Open-Systems-Adapter device.

**SENSEd**

tells CP to use the information returned from a sense ID device command to determine the characteristics of this device. If omitted, the default is SENSED.

### NOTSENSEd

tells CP not to use the information returned from a sense ID device command to determine the characteristics of this command.

### Usage Notes

1. Users should specify EQIDs for the following device classes in order to use them in live guest relocation:
  - a. CTCA
  - b. FCP
  - c. HiperSockets
  - d. OSA

While the system can generate EQIDs for these device classes, they are not guaranteed to extract all information necessary for equivalency, such as connection fabrics.

2. When setting an EQID for a CTCA, FCP, HiperSocket, or OSA, if one or more devices already exist with the specified EQID, they all must be of the same device type (CTCA, etc.). If the new RDEV is of a different class or type, you will receive message HCP048E and the RDEV will not be created.
3. When setting an EQID for any other Special class RDEV, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
4. The system cannot generate EQIDs for devices that do not support RCD CCW.

### Examples

Example 1:

To specify a channel-to-channel adapter at device number 0250, enter the following SET RDEVICE command:

```
set rdevice 0250 type ctca
```

Example 2:

To specify an Open-Systems-Adapter device at device number 0252, enter this SET RDEVICE command:

```
set rdevice 0252 type osa
```

Example 3:

To specify an Open-Systems-Adapter device at device number 0252 with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0252 eqid ABCDEFGH type osa
```

Example 4:

To specify an Open-Systems-Adapter device at device number 0252 and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0252 noeqid type osa
```

### Responses

After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

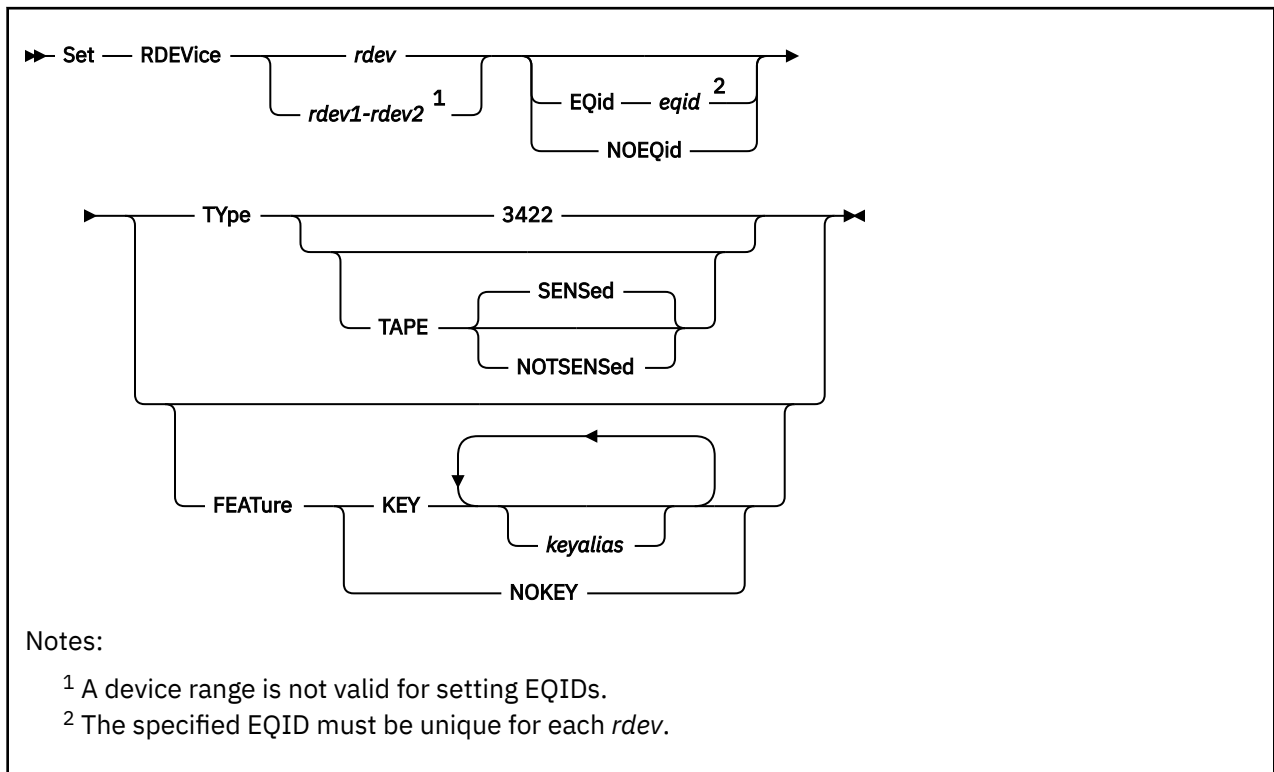
```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```



## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP048E Specified EQID already assigned to a different class of devices.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE Tape Units



### Authorization

Privilege Class: B

### Purpose

Use this SET RDEVICE command to tell CP that tape units are at certain device addresses.

### Operands

*rdev*

*rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

**Note:** A device range is not valid for setting EQIDs.

**EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters.

**Note:** For this device class, the specified EQID must be unique for each *rdev*.

**NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

**TType** 3422

specifies an IBM 3422 tape unit.

**TYpe TAPE**

specifies an IBM tape unit that can be dynamically sensed by CP (such as an IBM 3480 or 3490 tape unit).

**SENSEd**

tells CP to use the information returned from a sense ID request to determine the characteristics of this tape unit. If omitted, the default is SENSED.

**NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the characteristics of this tape unit.

**FEATure**

tells CP about features that an online, and free, tape device is capable of.

**KEY *keyalias***

prepares CP to enable a device's hardware encryption on behalf of a guest that is not able to manage the support itself. If a *keyalias* is specified (up to two are allowed), CP will use the encoding mechanism and key label described by the alias while setting up the encryption environment. If no aliases are supplied, the external key manager's default keys will be used.

**NOKEY**

indicates that CP should no longer try to manage the encryption environment on behalf of the guest.

**Usage Notes**

1. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
2. The system cannot generate EQIDs for devices that do not support RCD CCW.

**Examples**

Example 1:

To specify a 3422 tape unit at device number 0310, enter the following SET RDEVICE command:

```
set rdevice 0310 type 3422
```

Example 2:

To specify a tape unit that can be dynamically sensed by CP at device number 0315, enter this SET RDEVICE command:

```
set rdevice 0315 type tape
```

Example 3:

To specify a tape unit is capable of hardware encryption, but the guest will not be capable of using it itself, enter this SET RDEVICE command:

```
SET RDEVICE 07E2 FEATURE KEY
```

Example 4:

To instruct CP to stop managing the encryption settings for a tape unit, enter this SET RDEVICE command after detaching the tape from the guest:

```
SET RDEVICE 07E2 FEATURE NOKEY
```

Example 5:

## SET RDEVICE Tape Units

To specify a 3422 tape unit at device number 0252 with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0252 eqid ABCDEFGH type 3422
```

Example 6:

To specify a 3422 tape unit at device number 0252 and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0252 noeqid type 3422
```

## Responses

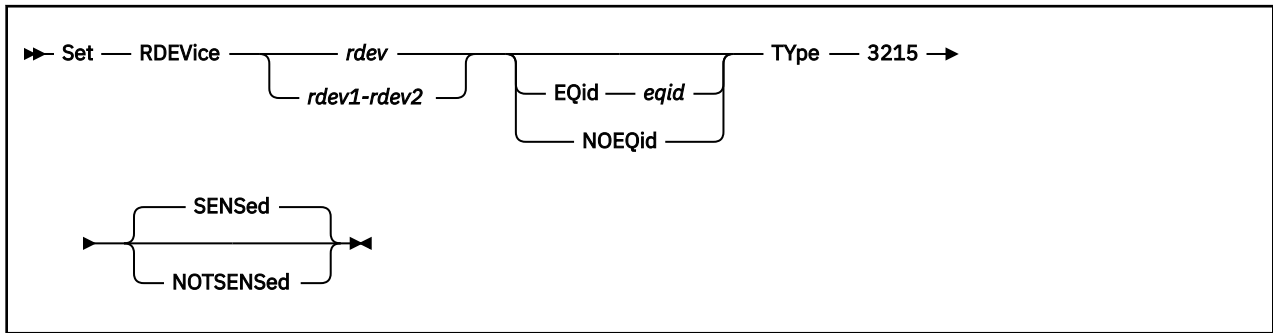
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP013E Conflicting option - *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP1550E *rdev* is an invalid device class.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*
- HCP6706E Invalid model number - *number*
- HCP6721E Device *rdev* must be free and online to respond to a SET RDEVICE FEATURE command.
- HCP6721E Tape device *rdev* must be at Beginning of Tape or unloaded to a SET RDEVICE FEATURE command.

## SET RDEVICE Terminals



### Authorization

Privilege Class: B

### Purpose

Use this SET RDEVICE command to tell CP that terminals are defined at certain device addresses.

### Operands

#### *rdev*

#### *rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

#### **EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

#### **NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

#### **TYpe 3215**

specifies a 3215 terminal.

#### **SENSEd**

tells CP to use the information returned from a sense ID device command to determine the characteristics of this device. If omitted, the default is SENSED.

#### **NOTSENSEd**

tells CP not to use the information returned from a sense ID device command to determine the characteristics of this command.

### Usage Notes

1. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCPO48E and the RDEV will not be created.
2. The system cannot generate EQIDs for devices that do not support RCD CCW.

### Examples

Example 1:

## SET RDEVICE Terminals

To specify a 3215 terminal at device number 0253, enter this SET RDEVICE command:

```
set rdevice 0253 type 3215
```

Example 2:

To specify a 3215 terminal at device number 0252 with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0252 eqid ABCDEFGH type 3215
```

Example 3:

To specify a 3215 terminal at device number 0252 and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0252 noeqid type 3215
```

## Responses

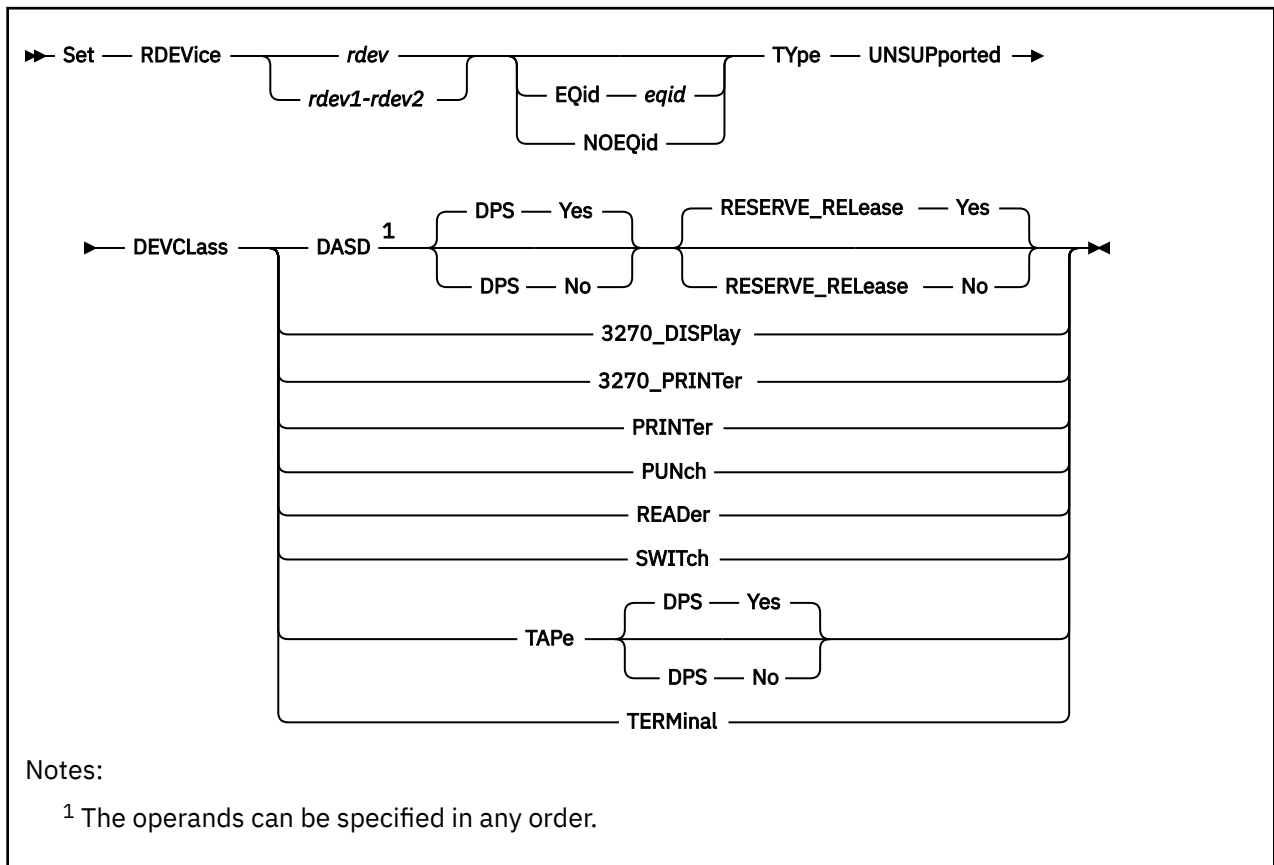
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE Unsupported Devices



### Authorization

Privilege Class: B

### Purpose

When you generate a device as an unsupported device, you must dedicate the device to a virtual machine. You can dedicate a device to a virtual machine by coding the DEDICATE control statement in the virtual machine's directory entry or by entering the CP ATTACH command.

### Operands

**rdev**

**rdev1-rdev2**

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

**EQid *eqid***

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

**NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

## SET RDEVICE Unsupported Devices

### **TType UNSUPported**

specifies an unsupported device type.

### **DEVClass**

is the device class of the unsupported device.

Valid classes for unsupported devices include the following:

<b><u>Class Value</u></b>	<b><u>Class for Unsupported Device</u></b>
DASD	Direct access storage devices
3270_DISPLAY	Graphics display devices
3270_PRINTEr	Graphics printer devices
PRINTEr	Unit record printer devices
PUNch	Unit record punch devices
READer	Unit record input devices
SWITch	Dynamic switching devices
TAPe	Tape devices
TERMinal	Terminals

### **DPS No**

### **DPS Yes**

The default is DPS YES. This indicates a DASD or TAPE device being defined supports the dynamic path selection (DPS) function.

DPS NO should only be specified for unsupported DASD and TAPE devices that do not support the DPS function.

### **RESERVE\_RELEASE No**

### **RESERVE\_RELEASE Yes**

The default is RESERVE\_RELEASE YES. This indicates the DASD being defined supports the reserve/release function.

RESERVE\_RELEASE NO should only be specified for DASD devices that do not support the reserve/release function.

## **Usage Notes**

1. The SET RDEVICE command can be used:
  - To define a device to CP for which no control blocks exist. CP will build the required control blocks to use this device.
  - To modify the control blocks for an existing device. In this case the DPS Yes parameter is not able to change the handling of the device by CP (from DPS NO to DPS YES). You must have the device defined in the SYSTEM CONFIG file with DPS YES and perform an IPL CLEAR of the entire system.
2. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
3. The system cannot generate EQIDs for devices that do not support RCD CCW.

## **Examples**

Example 1:

To specify an unsupported DASD, enter the following SET RDEVICE command:



```
set rdevice 0fff type unsupported devclass dasd
```

**Example 2:**

To specify an unsupported DASD that supports dynamic path selection, enter this SET RDEVICE command:

```
set rdevice 0fe0 type unsupported devclass dasd dps yes
```

**Example 3:**

To specify an unsupported DASD that supports both dynamic path selection and reserve/release, enter this SET RDEVICE command:

```
set rdevice 0fd0 type unsupported devclass
dasd dps yes reserve_release yes
```

**Example 4:**

To specify an unsupported DASD at device number 0fff with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0fff eqid ABCDEFGH unsupported devclass dasd
```

**Example 5:**

To specify an unsupported DASD at device number 0fff and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0fff noeqid unsupported devclass dasd
```

**Responses**

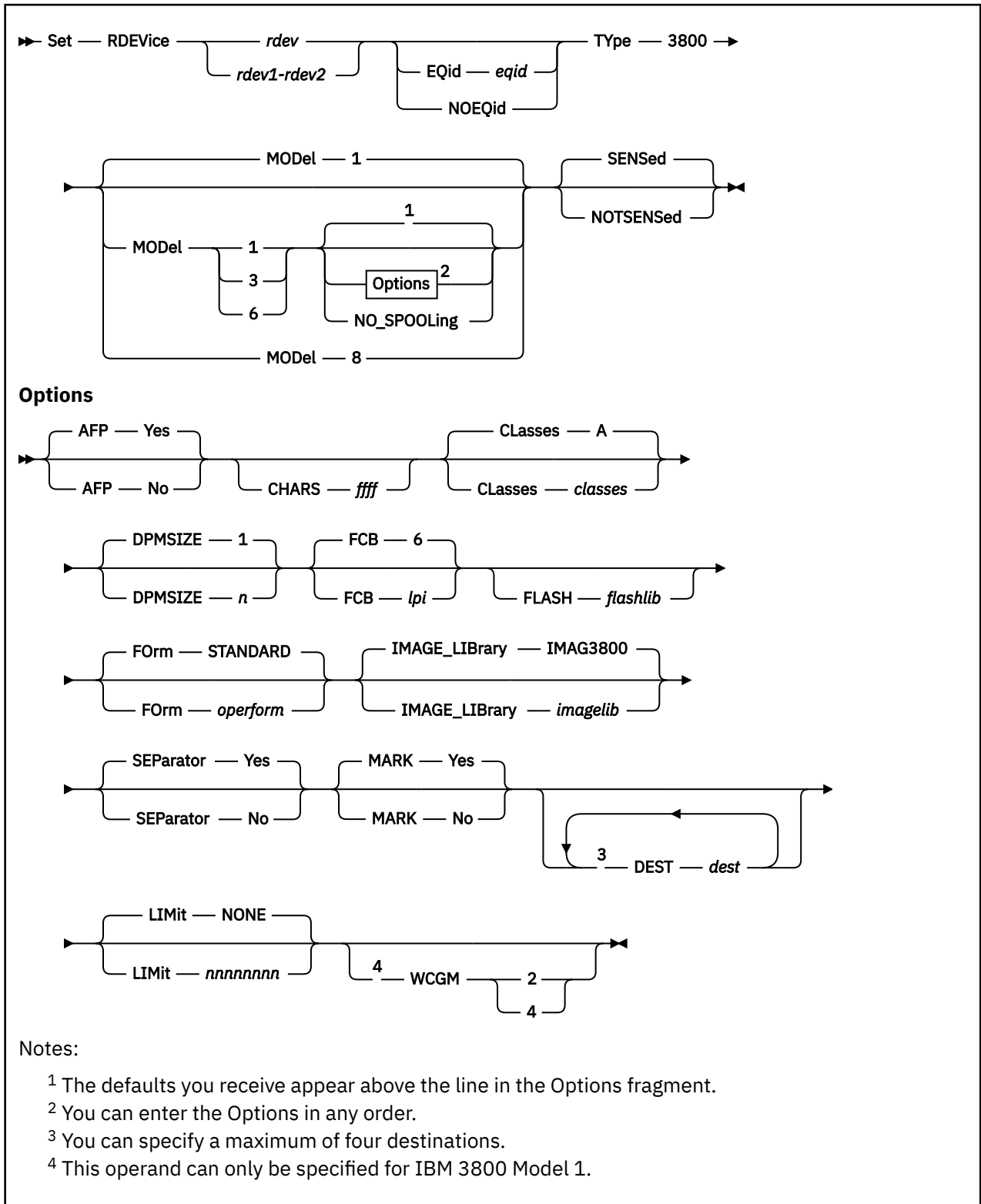
After CP has issued a message for each device address affected by a SET RDEVICE command, the following summary response is displayed:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP013E Conflicting option - *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*

## SET RDEVICE 3800 Printers



### Authorization

Privilege Class: B

## Purpose

Use this SET RDEVICE command to define or change 3800 printers.

## Operands

### *rdev*

#### *rdev1-rdev2*

specifies one or more real device numbers to be defined or changed. Each *rdev* specified must be a hexadecimal number between X'0000' and X'FFFF'. The maximum number of devices allowed within a range is 256.

### **EQid** *eqid*

assigns the device equivalency ID (EQID) *eqid* to the RDEV. The *eqid* is a string of 1–8 alphanumeric characters. Note that for this device class, the specified EQID must be unique across all devices on the system.

### **NOEQid**

removes a previously assigned EQID from this RDEV. After the device is varied online, it reverts back to a system-generated EQID.

### **TYpe 3800**

tells CP that this is a 3800 printer.

### **MODeI**

is the model number of the 3800 that you are defining.

#### **1**

is the default.

#### **3**

is supported either in IBM 3800 Model 1 compatibility mode, defined as a Model 1, or by using the Advanced Function Printing licensed program, defined as a Model 3. If you are using an IBM 3800 Model 3 as an advanced function printer, enter AFP YES.

#### **6**

is supported either in IBM 3800 Model 1 compatibility mode, defined as a Model 1, or by using the Advanced Function Printing licensed program, defined as a Model 6. If you are using a IBM 3800 Model 6 as an advanced function printer, enter AFP YES.

#### **8**

is supported using the Advanced Function Printing licensed program.

### **NO\_SPOOLing**

tells CP not to use the unit record device for printing.

### **SENSEd**

tells CP to use the information returned from a sense ID request to determine the characteristics of this printer. If omitted, the default is SENSED.

### **NOTSENSEd**

tells CP not to use the information returned from a sense ID request to determine the characteristics of this printer.

### **AFP Yes**

### **AFP No**

tells CP whether the printer will process files with advanced function printer (AFP) characteristics, extended attribute buffers (XABs), or X'5A' channel command words (CCWs). The default is YES.

### **CHARS** *ffff*

is the name of a character arrangement table for the separator page to use after a cold start. For 3800 printers, the default values are:

- For the IBM 3800 Model 1, CHARs GF10.
- For the IBM 3800 Model 3 or Model 6, CHARs GF12.

**Classes**

are the output spooling classes that this 3800 printer can print. (To change the spooling classes without having to take the printer offline, use the CP START command.)

**A**

is the default class.

**classes**

are the output spooling classes that the printer can print.

You can specify CLASSES in one of the following ways:

- Do not specify the CLASSES operand for unit record devices. Class A is the default.
- Enter CLASSES c[c...] to list up to 8 output spooling classes. The spooling class, c, is 1 alphanumeric character. You can specify multiple classes as CLASSES ABC and a single class as CLASSES A.
- Enter CLASSES \* to have the 3800 processes files regardless of their class.

**DPMSIZE n**

is the maximum size of the delayed purge queue (see note 1) that the 3800 Model 1, Model 3, or Model 6 printer uses after a cold start. The default value for n is 1, the maximum is 9. If '0' is specified, no delayed purge queue is maintained (see note 2).

**Notes:**

1. After the 3800 prints a file, CP places the file on the delayed purge queue if a delayed purge queue is being maintained. When the queue is full, CP purges the oldest file. This delay helps to ensure the 3800:
  - Transfers the last page of the file from the page buffer in the 3800 printer to paper
  - Stacks the printed page.If the 3800 fails before CP purges a file from the delayed purge queue, CP places the file in system hold.
2. Specifying '0' for DPMSIZE will save some spool DASD space because files printed on the 3800 will be purged immediately. However, this will decrease the possibility of recovering a printer file which has failed during printing.

**FCB lpi**

is the name of the forms control buffer for the separator page to use after a cold start. You can override this value by naming a forms control buffer on the START command. The default value for lpi is 6.

The number of lines to be printed on the separator page is determined by the FCB loaded on the printer. The default separator page contains 58 lines of data. If the page length defined by the FCB is less than the default separator page length, the separator page data must be customized in order to fit on a single page. This may be done using CP Exit points that are provided in separator page processing. See CP EXIT 4402: [Separator Page Perforation Printing or 3800 Positioning in z/VM: CP Exit Customization](#) for details.

**FLASH flashlib**

is the flash overlay to use for this device.

**FOrm operform**

is the current spooling form number that the printer can process. This form is the default operator form for the real printer. Specify this operand when defining a new printer or when redefining an existing printer. You can specify this operand in one of the following ways:

- Enter FORM operform, where operform is a 1- to 8-character operator form number for the files the printer can process.
- Enter FORM \* to have the printer process files regardless of form number.
- Enter FORM STANDARD to indicate the type of paper to which the installation has assigned the name STANDARD. Each installation establishes its own set of form names, assigns those form

names to types of paper, and tells the operations staff and end users which form name to use for which type of paper.

STANDARD is the default.

### **IMAGE\_LIB**rary *imagelib*

is the name of the image library to use after a cold start. You can override this value by specifying an image library name on the START command.

The default value is IMAGE IMAG3800.

### **SE**parator Yes

### **SE**parator No

tells CP whether to insert a separator page between output files. YES is the default.

### **MARK** Yes

### **MARK** No

specifies whether a 3800 printer will mark separator trailer pages with separator bars. When 'Mark Yes' is in effect, 3 separator trailer pages are printed for each file. 'Mark No' causes only one trailer page to be printed, thus saving paper. However, using 'Mark No' could also make it more difficult to separate 3800 output, as there are no markings between files. The default is YES.

### **DEST** *dest*

is the destination that an output device can service. The variable *dest* is a 1- to 8-character alphanumeric string assigned to the printer. You may specify up to four destination values.

### **LIM**it NONE

### **LIM**it *nnnnnnnnn*

is the maximum number of records of a file to be printed. The default is NONE.

### **WCGM** 2

### **WCGM** 4

tells CP that the 3800 has either the 2-writable-character generation module feature or the 4-writable-character generation module feature.

**Note:** You can only specify this option for a IBM 3800 Model 1.

For a IBM 3800 Model 1, the default is WCGM 2. For IBM 3800 Models 3, 6, and 8, the default is WCGM 4.

## **Usage Notes**

1. When setting an EQID for a device of this class, the specified EQID must be unique across all devices in the system. If the specified EQID already exists, you will receive message HCP048E and the RDEV will not be created.
2. The system cannot generate EQIDs for devices that do not support RCD CCW.

## **Examples**

Example 1:

To specify an IBM 3800 Model 3 printer at device number 500 to be dedicated to a virtual machine, enter the following:

```
set rdevice 500 type 3800 model 3
```

Example 2:

To specify three IBM 3800 Model 8 printers at device numbers 501, 502, and 503 to be dedicated to virtual machines, enter the following:

```
set rdevice 501-503 type 3800 model 8
```

Example 3:

## SET RDEVICE 3800 Printers

To specify an IBM 3800 Model 3 printer at device number 0252 with an EQID of ABCDEFGH, enter this SET RDEVICE command:

```
set rdevice 0252 eqid ABCDEFGH type 3800 model 3
```

Example 4:

To specify an IBM 3800 Model 3 printer at device number 0252 and remove any previously defined EQID, enter this SET RDEVICE command:

```
set rdevice 0252 noeqid type 3800 model 3
```

## Responses

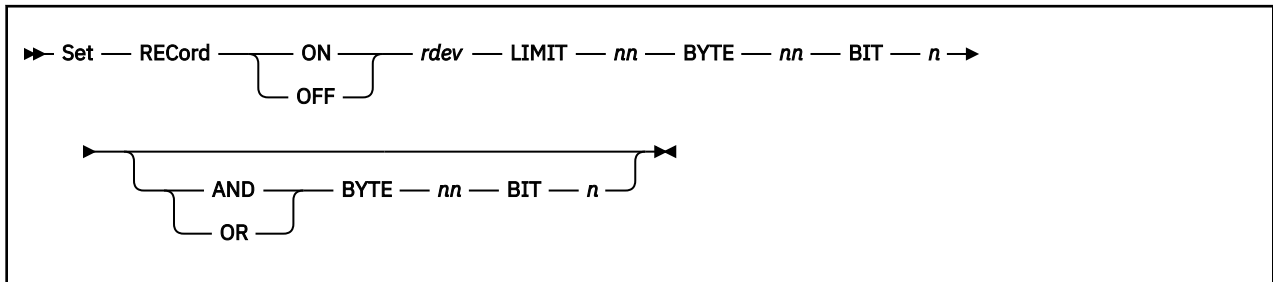
After CP issues a message for each device address affected by a SET RDEVICE command, CP displays the following summary response:

```
xxx RDEV(s) specified; yyy RDEV(s) changed; zzz RDEV(s) created
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP048E EQID must be unique for the specified device class.
- HCP048E Specified EQID already assigned to a different device.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *rdev-rdev*

## SET RECORD



### Authorization

Privilege Class: F

### Purpose

Use SET RECORD to set the recording mode for a device.

### Operands

#### ON

#### OFF

controls the intensive recording mode for the device that you specify. You can specify intensive recording for only one device at a time.

If you do not enter SET RECORD OFF, intensive recording is automatically terminated after 10 error records have been accumulated for the device, or when you activate intensive recording for a different device.

#### *rdev*

identifies the real device involved.

#### LIMIT *nn*

specifies the value at which a recording is made. While intensive recording is active, an error record is made for the first and for every *nn*th qualifying I/O error from the specified device.

#### BYTE *nn*

#### BIT *n*

the BYTE and BIT operands define the test condition in the sense bytes. You can test two bits using an AND or OR relationship. The range for BYTE is 0 through 31; the range for BIT is 0 through 7.

#### AND

#### OR

is the logical operation that you want to perform on two bits of sense data.

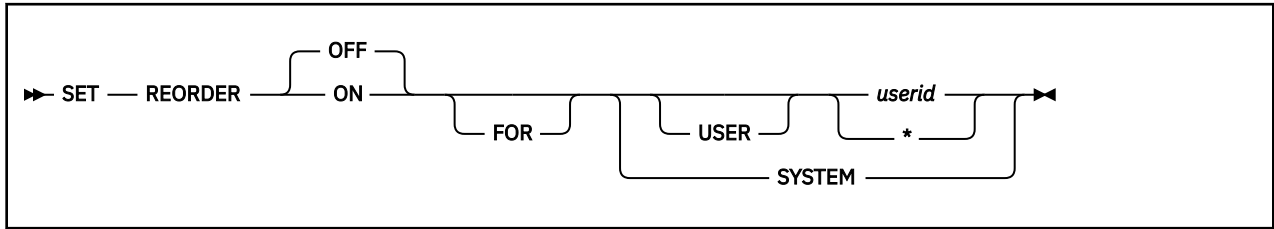
### Usage Notes

1. For more information on error recording, see [Collecting Hardware Error Information](#) in *z/VM: System Operation*.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev*/*vdev*/*ldev*} does not exist

## SET REORDER



### Authorization

Privilege Class: B

### Purpose

With the introduction of z/VM memory scalability up to 1T of real memory, the reorder process is no longer supported. The SET REORDER command remains for compatibility purposes. Issuance of SET REORDER ON results in message HCP6005E.

### Operands

#### ON

#### OFF

This function is no longer supported. The default is OFF.

#### FOR

#### USER

are optional keywords to aid in readability of the command.

#### *userid*

applies the command setting to the User Frame Owned list of the specified virtual machine.


#### \*

applies the command setting to the User Frame Owned list of the command issuer's virtual machine.

#### SYSTEM

applies the command to the User Frame Owned lists for all virtual machines and also to the Shared Frame Owned list.

### Usage Notes

- 
**Attention:** This command should be used only when it has been determined that a virtual machine's periodic performance slowdown is the result of the reorder task. You should consult with IBM support personnel before using this command to ascertain whether other system characteristics may be causing the poor performance. See [z/VM Reorder Processing](#) for more information.
- You can use the COMMAND directory statement to specify a reorder setting for a specific virtual machine. The command will be executed after the virtual machine logs on. Any subsequent SET REORDER command to change the system setting will apply to all virtual machines, including those having an individual override in their directory entry who are currently logged on.
- If you issue the SET REORDER command for a specific virtual machine and subsequently issue the SET REORDER command with the SYSTEM keyword, the SYSTEM setting will override the virtual machine's setting. You can, however, change the setting for a specific virtual machine after the SYSTEM setting is as desired.
- The current value of the SYSTEM setting applies to a virtual machine at logon.



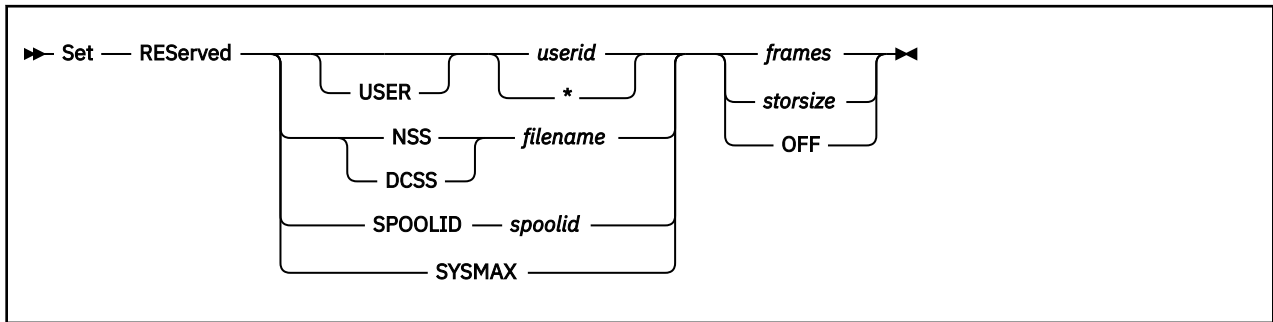
## Responses

Reorder is ON|OFF for user *userid*  
Reorder default is ON|OFF for SYSTEM

## Messages

- HCP002E Invalid operand - *operand*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E User not logged on
- HCP6005E Option *option* is not supported by z/VM.

## SET RESERVED



### Authorization

Privilege Class: A

### Purpose

Use SET RESERVED to establish the amount of storage a virtual machine, named saved system (NSS), or discontinuous saved segment (DCSS) is entitled to have resident under normal system operation. The system's frame replenishment task honors the reserved setting when searching for frames to take in order to replenish the available frame lists. CP will not take frames from a virtual machine, NSS, or DCSS if doing so would reduce its resident storage footprint below its reserved storage setting unless a system outage is imminent due to a severe shortage of frames.

This command may also be used to specify a maximum reserved storage setting. The total amount of reserved storage for all virtual machines, NSSs, and DCSSs cannot be set higher than the system maximum.

You should evaluate the storage requirements of your system workload in determining when best to use this command. Overuse can limit its effectiveness and negatively affect overall performance by limiting the frames that can be stolen to satisfy system needs.

### Operands

#### USER

an optional keyword you can use when specifying a virtual machine's user ID.

#### *userid*

\*

identifies the virtual machine. To specify your own virtual machine, enter an asterisk (\*).

#### NSS *filename*

identifies a system data file containing an available NSS.

#### DCSS *filename*

identifies a system data file containing an available DCSS.

#### SPOOLID *spoolid*

identifies a system data file containing an available or pending purge NSS or DCSS. Since multiple instances of an NSS or DCSS can be pending purge, you must identify an NSS or DCSS by spool id if the NSS or DCSS is in the pending purge state.

#### SYSMAX

defines the maximum amount of storage that can be reserved for all virtual machines, NSSs, and DCSSs. If no system maximum is in effect, the total amount of storage that can be reserved is limited to the current size of the dynamic paging area (DPA) at the time the SET RESERVED command is issued. It is recommended you explicitly set a system maximum (via the STORAGE RESERVED SYSMAX statement in the SYSTEM CONFIG file or by using the SET RESERVED SYSMAX command)

rather than rely on the DPA check. The size of the DPA is a floating value that is dependent on workload and CP behavior.

There is no range checking of the system maximum. The command response shows the specified system maximum as a percentage of the DPA and of online real storage. You need to be aware that reserving storage up to or near the size of the DPA will adversely affect system performance. Even when an explicit system maximum is in effect, you will not be able to increase the total amount of reserved storage beyond the size of the DPA at the time the SET RESERVED command is issued.

If your system has reconfigurable storage, it is recommended that you explicitly set the system maximum (via the STORAGE RESERVED SYSMAX statement in the SYSTEM CONFIG file or by using the SET RESERVED SYSMAX command) to a value that is less than the size of permanent storage. This is particularly important when you are planning to reclaim all memory from reconfigurable storage, to make sure the total reserved storage on the system does not exceed the size of permanent storage.

### **frames**

### **storsize**

### **OFF**

is the amount of storage to be reserved for a virtual machine, NSS, or DCSS or to be used as the system maximum. Specify OFF to reset a reserved setting to 0. In the case of SYSMAX, a setting of OFF reverts the system maximum to the current size of the DPA.

The amount of storage to be reserved can be identified by either:

- A 1- to 8-digit decimal count of frames. A frame count of 0 removes a reserved setting and is equivalent to specifying OFF.

Specification of the reserved setting as a count of frames is maintained for compatibility with earlier releases. The SET and QUERY RESERVED command responses always show the reserved value as a storage size even when the setting was originally specified as a count of frames.

- A storage size in format *nu* where *n* is a 1- to 7-digit decimal number and *u* is the 1-character storage unit suffix (M,G,T,P,E).

A storage size of 0 removes a reserved setting and is equivalent to specifying OFF.

## **Usage Notes**

1. To avoid confusion with the CMS SET RESERVED command, you should enter the SET RESERVED command with the CP prefix (for example, `cp set reserved * 10` or you should be in CP Read mode when you enter the command).
2. Unlike the CP LOCK command, SET RESERVED does not guarantee a specific resident page will remain resident, rather it is used to protect the amount of storage a guest, NSS, or DCSS has resident to assist in performance tuning.
3. When establishing a reserved storage setting for a user, the user must be logged on.
4. The maximum amount of storage you can reserve for a user is limited by the amount of private addressable storage in all private address spaces owned by the user's virtual machine. A shared address space is not protected by a user's reserved setting.
5. The maximum number of frames you can reserve for an NSS or DCSS is limited to the number of shared pages in the NSS or DCSS definition. Exclusive pages within an NSS or DCSS are counted as private addressable storage owned by the virtual machine.
6. The DCSS must not be a member segment (filetype DCSS-M) or a segment space (filetype DCSS-S).
7. When an NSS or DCSS with reserved storage is made pending purge (class P), the reserved setting remains until the system data file is actually purged (when it is no longer loaded by any user). Any changes to the reserved setting for an NSS or DCSS in the pending purge state must be done using the SPOOLID form of the SET RESERVED command. You can use the QUERY NSS NAME command to obtain the spoolid for the NSS or DCSS.
8. If a new instance of an NSS or DCSS is created (same filename) and the previous instance had a reserved setting, the reserved setting is not automatically inherited by the new instance. A new SET

## SET RESERVED

RESERVED command must be issued to establish reserved storage for the new instance of the NSS or DCSS.

9. If you specify a value for SYSMAX that is less than the total reserved storage already established, the new SYSMAX value prevents any new reserved storage allocations, but the existing allocations are not affected. Once the total amount of reserved storage drops below the SYSMAX (for example, when a guest with reserved storage logs off), additional reserved storage allocations will be allowed.
10. Increasing the total amount of reserved storage in the system is always subject to the size of DPA at the time the SET RESERVED command is issued, even when the SYSMAX in effect exceeds the size of DPA.

When the total amount of reserved storage is decreased, the new total is compared to the current SYSMAX setting and the size of DPA. If the new total exceeds either SYSMAX or the DPA size, message HCP2548I is issued to inform you the total is set too high. In this case you should further reduce the individual reserved settings for selected virtual machines, NSSs or DCSSs.

11. The frame count specification on the SET RESERVED command is maintained for compatibility. The command response always displays the reserved setting as a storage size.
12. For readability, the storage values displayed in the command response are displayed with the same storage unit. The values for the resident and instantiated storage may be rounded to accommodate the unit specification of the reserved setting. The reserved setting will never be rounded although it may be shown in a smaller unit than entered, when the resident or instantiated storage size is less than the reserved setting, in order to maintain the same storage unit across the line.

If you want precise values for the resident and instantiated counts, you should use the INDICATE USER or INDICATE NSS commands.

13. Reserved settings are not preserved across an IPL.
14. When an NSS/DCSS is removed from storage because it is no longer in use, all of the NSS/DCSS pages are released. The reserved setting is preserved and will be reestablished as pages of the NSS/DCSS are made resident should the file be loaded again.

## Responses

Response 1:

Resource	Type	SPID	Reserved	Resident	Instantiated
userid	USER	----	rsvstorU	resstorU	instanstorU

where:

### **userid**

identifies the virtual machine.

### **USER**

identifies the type of resource as a virtual machine.

----

four dashes are displayed to indicate a spool id is not applicable when Type=USER.

### **rsvstorU**

is the reserved storage setting for this userid in storage units (K,M,G,T,P,E).

### **resstorU**

is the resident storage size for this userid in storage units (K,M,G,T,P,E) at the time the command is issued. This value may be rounded.

### **instanstorU**

is the instantiated storage size for this userid in storage units (K,M,G,T,P,E) at the time the command is issued. This value may be rounded. Since guest pages can exist in multiple levels of the storage hierarchy (for example, in real storage and in auxiliary storage), the instantiated storage may be less than the storage size calculated by summing the guest's page counts for real and auxiliary storage as shown in response to the INDICATE USER command.

Response 2:

Resource	Type	SPID	Reserved	Resident	Instantiated
filename	type	SPID	rsvstorU	resstorU	instanstorU

where:

**filename**

identifies the NSS or DCSS.

**type**

identifies the type of resource as NSS, DCSS, or DCSSG.

**spid**

is the 4-digit spool id associated with an NSS or DCSS. If the file is pending purge, the letter P is appended to the spool id.

**rsvstorU**

is the reserved storage setting for this userid in storage units (K,M,G,T,P,E).

If the reserved value was specified as count of frames on the SET RESERVED command, the value may be rounded to fit within 7-decimal digits plus a storage unit suffix.

**resstorU**

is the resident storage size for this userid in storage units (K,M,G,T,P,E) at the time the command is issued. This value may be rounded.

**instanstorU**

is the instantiated storage size for this userid in storage units (K,M,G,T,P,E) at the time the command is issued. This value may be rounded. Since guest pages can exist in multiple levels of the storage hierarchy (for example, in real storage and in auxiliary storage), the instantiated storage may be less than the storage size calculated by summing the guest's page counts for real and auxiliary storage as shown in response to the INDICATE USER command.

Response 3: Upon successful completion of the SET RESERVED SYSMAX command, the following command response is displayed:

```
SYSMAX = nU    xxx% of pageable storage    yyy% of online storage
```

where:

**nU**

identifies the total maximum allowable amount of reserved storage for all users, NSSs, and DCSSs in storage units (K,M,G,T,P,E).

**xxx%**

specifies the system maximum as an approximate percentage of the Dynamic Paging Area (DPA). The percent value is rounded to the nearest whole percent. When the value is less than .5%, it is shown as <1%.

The size of the DPA varies over time. It includes available frames and those that are pageable at the time the command is issued. A system maximum near the size of DPA can have undesirable effects on performance.

**yyy%**

specifies the system maximum as an approximate percentage of real online storage. The percent value is rounded to the nearest whole percent. When the value is less than .5%, it is shown as <1%.

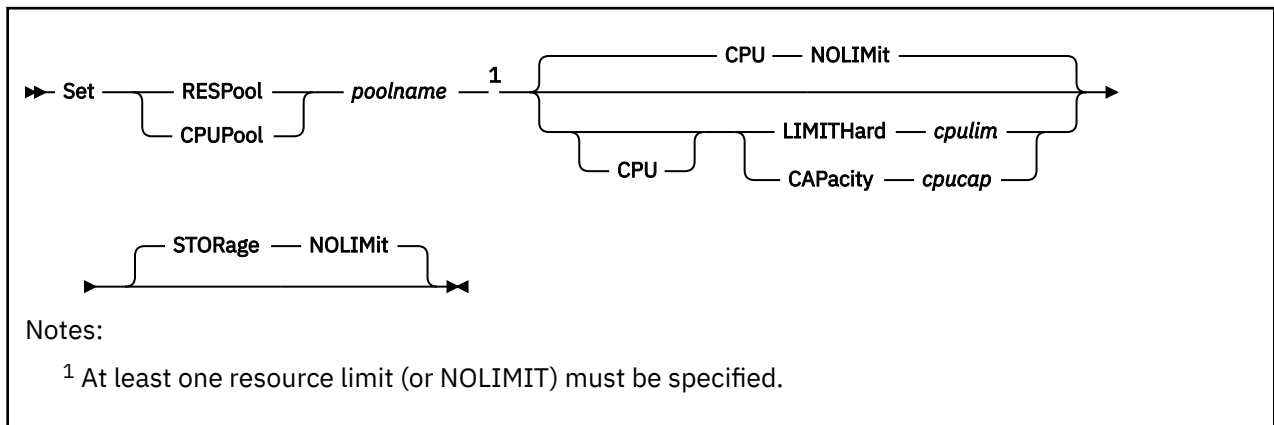
Response 4: When the reserved setting is set to OFF (or to zero), the following response is displayed:

```
RESERVED is OFF for userid|filename|SYSMAX
OR
RESERVED is OFF for spoolid spoolid
```

**Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- SET RESERVED was not processed. Filename *filename* does not identify {an NSS | a DCSS} in an available state.
- HCP2547E SET RESERVED was not processed. Spoolid *spoolid* does not identify an NSS or DCSS in an available or pending purge state.
- HCP2547E SET RESERVED was not processed. The DCSS with {filename *filename* | spoolid *spoolid*} is a member segment or segment space.
- HCP2547E SET RESERVED was not processed. *rsvsize* exceeds the amount of private addressable storage (*privsize*) for user *userid*.
- SET RESERVED was not processed. *rsvsize* exceeds the amount of shared storage (*sharsize*) for the {NSS | DCSS} with filename *filename*.
- SET RESERVED was not processed. *rsvsize* exceeds the amount of shared storage (*sharsize*) for the NSS or DCSS with spoolid *spoolid*.
- HCP2547E SET RESERVED was not processed. Total reserved would exceed the system maximum of *maxsize* by *storsize*.
- HCP2547E SET RESERVED was not processed. Total reserved would exceed the current DPA by *storsize*.
- HCP2548I SET RESERVED completed successfully. However, total reserved still exceeds the system maximum of *maxsize* by *storsize*.
- HCP2548I SET RESERVED completed successfully. However, total reserved still exceeds the current DPA by *storsize*.
- HCP6704E Missing token at end of line

## SET RESPOOL



### Authorization

Privilege Class: A

### Purpose

Use SET RESPOOL to change the CPU or storage limit setting for a resource pool. For compatibility with prior releases, CPUPOOL is accepted as a synonym for RESPOOL.

### Operands

#### *poolname*

is the name of the resource pool whose CPU or storage limit setting is to be changed.

#### **CPU LIMITHard *cpulim***

#### **LIMITHard *cpulim***

sets the enforcement of hard CPU limiting for the defined pool such that it does not receive more than *cpulim* percent of the shared logical cores of the type specified on the TYPE operand when the pool was defined. Here, "logical core" refers to a logical CPU when multithreading is not enabled, or to a set of logical processors (threads) currently dispatched on a physical processor core when multithreading is enabled. The *cpulim* value is a decimal whole number (no decimal place) from 1 to 100, followed by a percent sign (for example, 2% or 80%).

#### **CPU CAPacity *cpucap***

#### **CAPacity *cpucap***

sets the enforcement of CPU capacity limiting to the defined pool such that it does not receive an amount of processor resource greater than the equivalent of *cpucap* real cores of the type specified on the TYPE operand when the pool was defined (although the number of logical cores of that type in use simultaneously by the group can exceed *cpucap*). The *cpucap* value is a decimal number (up to two decimal places) from 0.01 to 999 (for example, 2.5 or 8).

#### **CPU NOLIMIT**

clears the enforcement of CPU capacity limiting for the defined pool.

#### **STORage NOLIMIT**

clears the enforcement of memory limiting for the defined pool.

### Usage Notes

1. The SET RESPOOL command can be issued only for a resource pool that has been created with the DEFINE RESPOOL command. For information on the types of CPUs that can be limited, see ["DEFINE RESPOOL"](#) on page 205).
2. Use the SCHEDULE command to assign users to or remove users from the resource pool.

## SET RESPOOL

3. LIMITHARD 100% is equivalent to and reported as CPU NOLIMIT.

### Responses

Response 1:

```
Resource pool poolname is changed
```

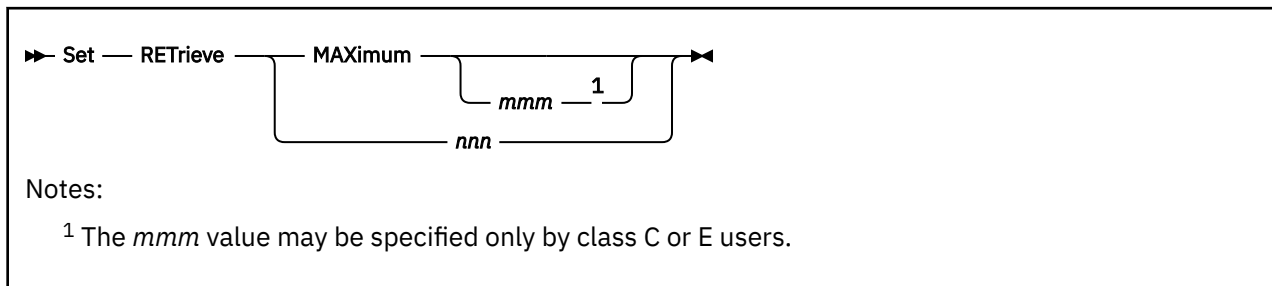
confirms that the resource pool has been changed as requested.

### Messages

- HCP002E Invalid operand *operand*
- HCP003E Invalid option – command contains extra option(s) starting with *option*
- HCP013E Conflicting option - *option*
- HCP1001E An operand is missing for CPU
- HCP1001E An operand is missing for LIMITHARD
- HCP1001E An operand is missing for CAPACITY
- HCP1001E An operand is missing for RESPOOL
- HCP1001E An operand is missing for STORAGE
- HCP3151E Resource pool *poolname* does not exist



## SET RETRIEVE



### Authorization

Privilege Class: C, E, G

### Purpose

Use SET RETRIEVE to allocate a specific number of program function key retrieve buffers for your virtual machine and to change the maximum number of program function key retrieve buffers that users may request.

### Operands

#### MAXimum

sets the number of program function key retrieve buffers available to your virtual machine to the maximum allowed. To set the maximum allowed, you can either use the class C and E SET RETRIEVE command or enter the information in the system configuration file using the FEATURES statement.

#### *mmm*

sets the maximum number of program function key retrieve buffers available to any virtual machine on the system to *mmm*. The value of *mmm* cannot exceed 255.

#### *nnn*

sets the number of program function key retrieve buffers available to your virtual machine to *nnn*. The value of *nnn* cannot exceed the previously defined maximum, which was set either with the FEATURES statement in the system configuration file or with the privileged version of the SET RETRIEVE command.

### Usage Notes

1. Reducing the maximum value *mmm* for program function key retrieve buffers does not affect the current program function key retrieve definitions for any user. Retrieve buffers are not deleted from users that exceed any newly established maximum.
2. After a Class C or Class E user has increased the maximum possible number of retrieve buffers, users do not have to log off to increase the number of program function key retrieve buffers available to their own virtual machines. A new SET RETRIEVE command can be used to increase the number of program function key retrieve buffers without affecting the contents of the existing buffers.
3. For more information on displaying the current number of program function key retrieve buffers available to your virtual machine and the maximum number of buffers that any user may request, see the QUERY RETRIEVE command.

### Examples

If you issue a SET PF $n$  with no parameters (for the PF key you had assigned to RETRIEVE), the number of buffers is set to zero, making your RETRIEVE ineffective.

## SET RETRIEVE

```
q pf1
PF01 RETRIEVE BACKWARD
q retrieve
7 buffers available. Maximum of 20 buffers may be selected.
Ready; T=0.01/0.01 16:06:56
set pf1
Ready; T=0.01/0.01 16:07:04
q retrieve
0 buffers available. Maximum of 20 buffers may be selected.
Ready; T=0.01/0.01 16:07:08
```

If you have two pf keys set for RETRIEVE, for example PF1 and PF12 and you set PF12 to undefined:

```
q pf1
PF01 RETRIEVE BACKWARD
Ready; T=0.01/0.01 16:18:52
q pf12
PF12 RETRIEVE BACKWARD
Ready; T=0.01/0.01 16:18:58
q retrieve
7 buffers available. Maximum of 20 buffers may be selected.
Ready; T=0.01/0.01 16:19:08
set pf12
Ready; T=0.01/0.01 16:19:15
q retrieve
7 buffers available. Maximum of 20 buffers may be selected.
Ready; T=0.01/0.01 16:19:20
```

then it does NOT set the buffer to zero.

SET RETRIEVE works to set a number of buffers for you even if you have no pf keys set to retrieve. However, if you then set your all your pf keys to null, the retrieve buffer is set to zero. Suppose you have only two pf keys set, PF3 and PF5:

```
q pf
PF01 UNDEFINED
PF02 UNDEFINED
PF03 IMMED INFOLIST
PF04 UNDEFINED
PF05 IMMED EXEC MAILBOX
PF06 UNDEFINED
.
PF24 UNDEFINED
Ready; T=0.01/0.01 17:05:03
```

and you set retrieve but then set PF3 and PF5 to null:

```
set retrieve 12
Ready; T=0.01/0.01 17:01:10
q retrieve
12 buffers available. Maximum of 20 buffers may be selected.
Ready; T=0.01/0.01 17:01:14
set pf3
Ready; T=0.01/0.01 17:01:25
set pf5
Ready; T=0.01/0.01 17:01:29
q retrieve
0 buffers available. Maximum of 20 buffers may be selected.
Ready; T=0.01/0.01 17:01:33
```

Your buffer is set to zero.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP6704E Missing token at end of line

## SET RUN



### Authorization

Privilege Class: G

### Purpose

Use SET RUN to control whether your virtual machine runs while a CP-initiated read is outstanding at your terminal.

### Operands

#### ON

#### OFF

When RUN is set ON, the following differences in virtual machine management occur:

1. If your virtual machine is runnable, it processes instructions even if a CP-initiated read is outstanding at your terminal
2. CP does *not* automatically begin another terminal read when an outstanding read is satisfied
3. If none of your configuration virtual processors are runnable, CP does *not* automatically begin a terminal read.

Initially, RUN is set OFF for all users except the primary system operator.

RUN may not be set ON after a trace trap has been established in an active trace set unless the trace set is suspended. This ensures that the virtual machine is not running during the terminal read that may result when a trace trap springs. Similarly, RUN is automatically set OFF when a trace trap is established in a non-suspended trace set.

### Usage Notes

1. An outstanding CP-initiated read does *not* necessarily indicate that your virtual machine is not executing instructions. When RUN is set ON, your virtual machine may process even with a CP-initiated read outstanding.
2. The SET RUN command allows the operator of a second-level system operating in full-screen mode to press the break key and enter CP commands without unnecessarily delaying the instruction execution of users of that second-level system. Note that when your terminal is operating in line mode, the same effect can be achieved by entering CP commands preceded by #CP, where # represents your logical-line-end character.
3. If your virtual machine is not runnable (for example, it is in a stopped or disabled-wait state), and RUN is set OFF, CP initiates a terminal read. On locally connected terminals and logical devices, this prevents CP from writing messages sent to the user ID. In this situation, RUN can be set ON to prevent CP from initiating the read; this keeps the terminal free for incoming messages.
4. Some CP commands prevent the virtual machine from running until all terminal output associated with the command has been displayed. If the output is delayed, for example, because the screen enters "MORE..." or "HOLDING" state, then the virtual machine will not run. The SET RUN ON command does not affect this behavior.

## SET RUN

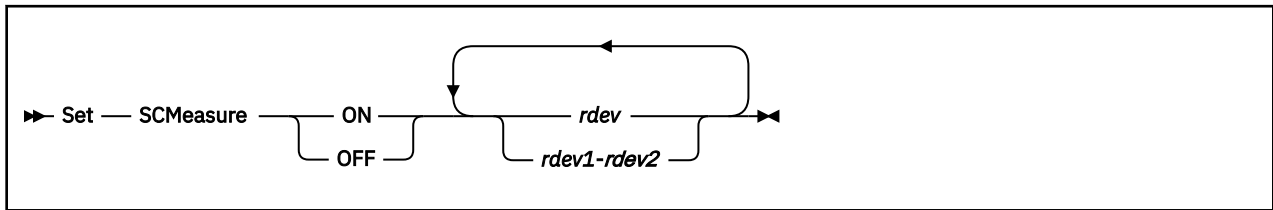
Similarly, virtual-machine I/O to the console will be queued during "MORE..." or "HOLDING" state, and the I/O completion will not be reported to the virtual machine until that state is cleared. If the virtual machine voluntarily enters wait state until its console I/O completes, then it will appear unresponsive until the "MORE..." or "HOLDING" condition is cleared.

5. In some cases, SET RUN ON may not enable a user to keep running during a CP-initiated read, or a similar situation. For example, if a console I/O error occurs, the user will not continue to run, and will eventually be FORCED off the system if a DISCONNECT\_TIMEOUT interval is set in the SYSTEM CONFIG file.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP1031I SET RUN ON is not allowed when the trace function is active.

## SET SCMEASURE



### Authorization

Privilege Class: B, E

### Purpose

Use SET SCMEASURE to enable or disable the collecting of subchannel measurement data for a specific device or range of devices.

### Operands

#### ON

tells CP to begin collecting subchannel measurement data for the specified device or devices.

#### OFF

tells CP to stop collecting subchannel measurement data for the specified device or devices.

#### *rdev*

#### *rdev1-rdev2*

is the real device number (or numbers) of the device on which you are enabling or disabling subchannel measuring. Each *rdev* must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single device address, a range of device addresses, or any combination thereof.

**Note:** You cannot specify a device that is eligible for I/O assist.

### Usage Notes

**Format-1 measurement blocks:** When format-1 measurement blocks are being used, measurement blocks are allocated dynamically.

1. Subchannel measurement blocks (SCMBKs) are used to accumulate data about a subchannel, such as rate and utilization information. This subchannel measurement data, which CP Monitor collects in various monitor records, can be useful for analyzing the I/O performance characteristics of a real device.
2. To display whether CP is collecting subchannel measurement data for one or more devices, use the QUERY SCMEASURE command (see [“QUERY SCMEASURE”](#) on page 1149).
3. When a device is first initialized, the default is to enable it for collecting subchannel measurements. To change this default for a given device, use the SCMEASURED or NOTSCMEASURED operands of the DEVICES statement in the system configuration file. For more information about the [DEVICES Statement](#), see *z/VM: CP Planning and Administration*.
4. The SCMEASURED and NOTSCMEASURED operands of the DEVICES statement define the default to be set for a device when it is initialized. You can use the SET SCMEASURE command at any time to change the current setting for a device. That is, if you use the DEVICES statement to indicate that a device should be disabled for collecting subchannel measurements when it is initialized, CP does not prevent you from turning subchannel measuring on for that device after it is initialized.

- The SET SCMEASURE command will assign an SCMBK to a device or return an assigned SCMBK to the available pool to be reused. To obtain information on the number of SCMBKs on your system, use the QUERY SCMBKS command (see [“QUERY SCMBKS”](#) on page 1147).

Format-0 measurement blocks:

- During initialization, CP allocates 1 SCMBK for every subchannel on the system and rounds the amount of storage up to the nearest page boundary. Thus, you have enough SCMBK space for every real device on your system plus some free SCMBK space. How much free SCMBK space you have will vary, depending upon how many real devices you have on your system.
- You should plan for additional SCMBK space if you intend to dynamically add new real devices to your running system. At initialization, you can allocate more free SCMBK space than CP would normally allocate using the SCMBK operand of the STORAGE statement in the system configuration file. For more information, see the [STORAGE Statement](#) in *z/VM: CP Planning and Administration*.
- If you run out of free SCMBK space on a running system and you need to measure additional real devices, you have two choices:
  - Re-IPL and specify more free SCMBK space using the SCMBK operand mentioned in Usage Note [“2”](#) on page 1686, or
  - Stop collecting subchannel measurement data for one or more devices. When you use one of the following commands:
    - DELETE DEVICE (see [“DELETE DEVICE / IODEVICE”](#) on page 265)
    - DELETE IODEVICE (see [“DELETE DEVICE / IODEVICE”](#) on page 265)
    - DELETE RDEVICE (see [“DELETE RDEVICE”](#) on page 274)
    - SET SCMEASURE OFF (see [“SET SCMEASURE”](#) on page 1685)

CP returns the SCMBK assigned to that device (or devices) to the pool of available SCMBKs. It is then available to be used to collect subchannel measurement data for another device.

## Responses

Response 1:

To tell CP to begin collecting subchannel measurement data for devices 127 through 130, enter the following SET SCMEASURE command:

```
set scmeasure on 127-130
Subchannel measuring for device 0127 is on
Subchannel measuring for device 0128 is on
Subchannel measuring for device 0129 is already on
Subchannel measuring for device 012A is already on
Subchannel measuring for device 012B is on
Subchannel measuring for device 012C is on
Subchannel measuring for device 012D is on
Subchannel measuring for device 012E is already on
Subchannel measuring for device 012F is already on
Subchannel measuring for device 0130 is on
Ready;
```

Note that CP was already collecting subchannel measurement data for devices 129, 12A, 12E, and 12F.

Response 2:

To disable subchannel measuring for devices 399 through 402, enter the following SET SCMEASURE command:

```
set scmeasure off 399-402
Subchannel measuring for device 0399 is off
Subchannel measuring for device 039A is off
Subchannel measuring for device 039B is off
Subchannel measuring for device 039C is already off
Subchannel measuring for device 039D is already off
Subchannel measuring for device 039E is off
Subchannel measuring for device 039F is already off
```

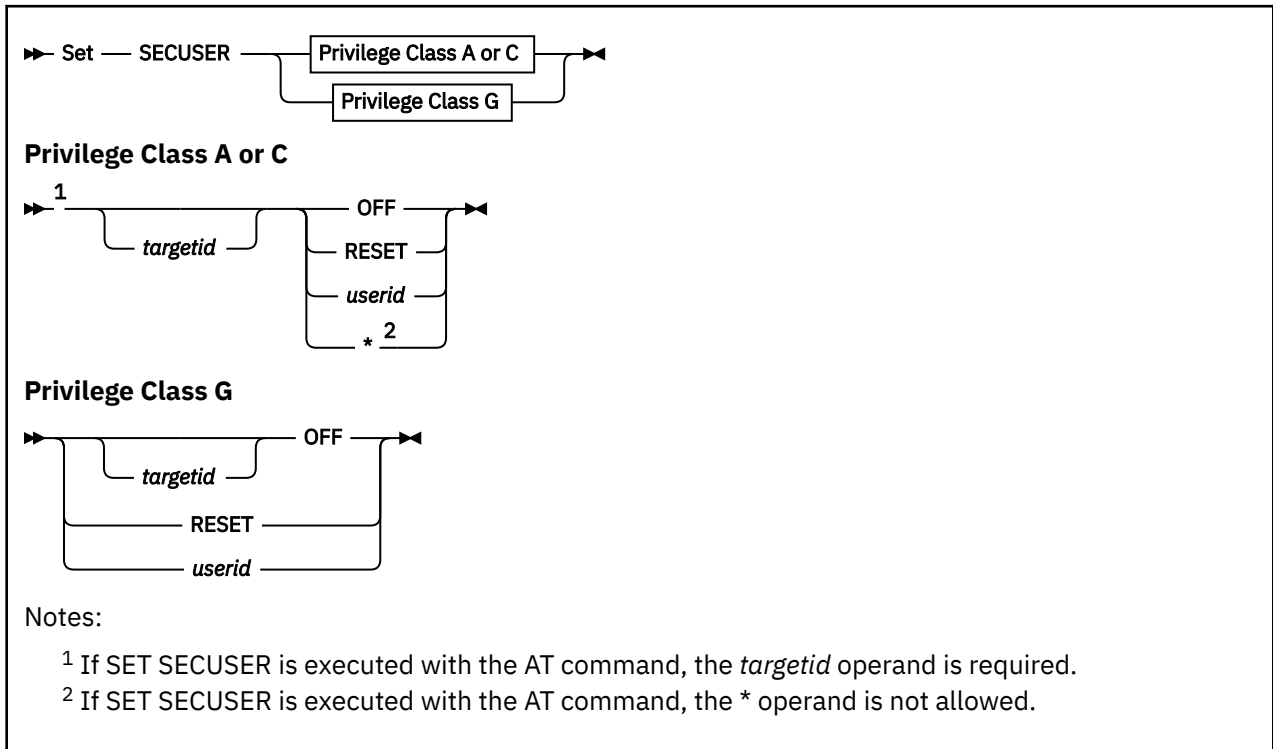
```
Subchannel measuring for device 0400 is off  
Subchannel measuring for device 0401 is already off  
Subchannel measuring for device 0402 is already off  
Ready;
```

Note that CP was not collecting subchannel measurement data for devices 39C, 39D, 39F, 401, and 402.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP040E Device *rdev* does not exist
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6704E Missing token at end of line
- HCP6706E Invalid address range - *range*
- HCP6706E Invalid RDEV - *rdev*. RDEV not initialized
- HCP6709E Too many items specified - *item*
- HCP6855E Command cannot be processed for device *rdev* because the subchannel cannot be modified
- HCP9423E Insufficient storage to allocate an SCMBK. Subchannel measurement will not be enabled for device *rdev*

## SET SECUSER



### Authorization

Privilege Class: A, C, G

### Purpose

Use SET SECUSER to change the secondary user for a virtual machine. The second user is a virtual machine that receives console messages from and can enter commands to the console of a disconnected virtual machine. If you are a class G user, you can change the secondary user for your own virtual machine. If you are a class A or C user, you can change the secondary user for any virtual machine.

### Operands

#### *targetid*

is the user ID of the virtual machine whose secondary user is to be set or reset. If you are a class A or C user and you do not specify *targetid*, CP sets or resets the secondary user for your user ID. If you are a class G user, you can stop being the secondary user for *targetid* by specifying *targetid* OFF.

#### OFF

cancels the secondary user setting. OFF terminates any console sharing with an existing secondary user.

#### RESET

sets the secondary user for *targetid* or the command issuer to the user ID specified on the CONSOLE statement in the user's directory entry.

#### *userid*

is the user ID of the secondary user virtual machine.

#### \*

sets the secondary user to the virtual machine that issues the command.



## Usage Notes

1. If secondary user support is used with large console messages that are close to the free storage limit of X'1FD' doublewords, it is possible that the message could be truncated at the end in order to add the header at the beginning which indicates the user ID from which the message originated.
2. A virtual machine cannot have both a secondary user and an observer defined by any combination of the SET SECUSER and SET OBSERVER commands and the CONSOLE directory statement.
3. If the designated secondary user is currently unable to function in that capacity (for example, not logged on, or disconnected with no connection to \*MSG or \*MSGALL), the setting is nevertheless accepted by CP, and the relationship is established when the specified user ID meets the necessary criteria.
4. In an SSI cluster, the secondary user and the primary user can be logged on to different member systems.

If either user is a multiconfiguration virtual machine:

- The secondary user will function in that capacity only when it is local (logged on to the same member as the primary user). If the primary user is a multiconfiguration virtual machine, the secondary user must be logged on to the member where the SET SECUSER command was issued.
- If the secondary user is remote:
  - The secondary user can issue SEND commands to the primary user. The AT *sysname* operands are required if the primary user is a multiconfiguration virtual machine instance.
  - The secondary user will not receive responses to SEND commands or any other output from the primary user.

## Examples

Example 1:

USERA (a general user) designates USERB (another general user) as its secondary user by issuing the following command:

```
set secuser userb
```

USERB can stop being the secondary user for USERA by issuing the following command:

```
set secuser usera off
```

If USERC is the secondary user for USERD, USERD can remove its secondary user by issuing the following command:

```
set secuser off
```

Example 2:

USERC (a class A or class C user) issues the following command to assign USERD as the secondary user for USERB:

```
set secuser userb userd
```

USERD receives the following message:

```
HCP6768I SECUSER of USERB initiated for you by USERC.
```

At the same time, USERB receives one of the following messages:

```
HCP6768I Your SECUSER set to USERD by USERC.
```

-OR-

## SET SECUSER

```
HCP6769I Your SECUSER terminated by USERC.
```

Example 3:

USERC can remove the secondary user for USERB by issuing the following command:

```
set secuser userb off
```

This causes the user ID that was the secondary user (USERD) to receive the message:

```
HCP6769I SECUSER of USERB terminated for you by USERC.
```

Example 4:

If the command issuer becomes the secondary user, the message issued at initiation is:

```
HCP6768I SECUSER of primary_userid initiated.
```

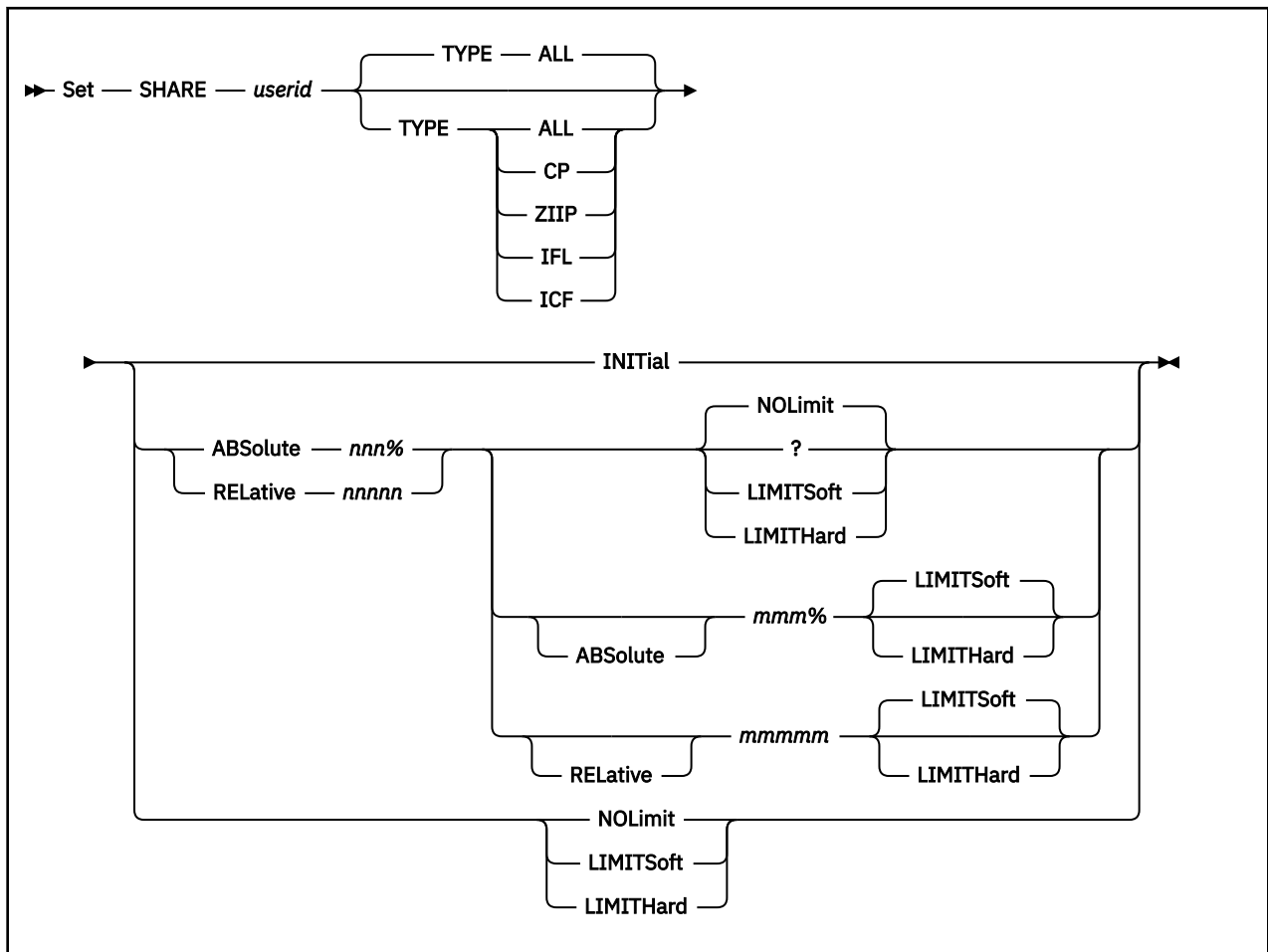
When the secondary user option is terminated, the message issued is:

```
HCP6769I SECUSER of primary_userid terminated.
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) beginning with *option*
- HCP007E Invalid userid - *userid*
- HCP017E User *userid* is being observed and cannot have a SECUSER
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP053E *userid* not in CP directory
- HCP0150E User *userid* has issued a CP or VM read.
- HCP1149E Error communicating with another SSI member. It cannot be determined if *command* executed remotely.
- HCP6768I SECUSER for *primary\_userid* of *secondary\_userid* is not functional
- HCP6768I SECUSER of *primary\_userid* initiated.
- HCP6768I SECUSER of *primary\_userid* initiated for you by *userid*.
- HCP6768I Your SECUSER set to *secondary\_userid* by *userid*.
- HCP6769I SECUSER of *primary\_userid* terminated.
- HCP6769I SECUSER of *primary\_userid* terminated for you by *userid*.
- HCP6769I Your SECUSER terminated by *userid*.

## SET SHARE



### Authorization

Privilege Class: A

### Purpose

Use SET SHARE to change the system-resource-access priority for users.

### Operands

#### *userid*

identifies the virtual machine whose share you are changing.

#### **ABSolute** *nnn%*

specifies that this user is to receive a target minimum of *nnn%* of the scheduled system resources, which include CPU, storage, and paging capacity; *nnn* is a decimal real number—no or one decimal place—from 0.1 to 100 (for example, 20.5% or 80%).

#### **RELative** *nnnnn*

specifies that this user is to receive a target minimum relative share of *nnnnn*. The amount of scheduled system resources available to relative share users is the total of resources available less the amount allocated to absolute share users. The portion that this user receives is *nnnnn* divided by the sum of the *nnnnn*'s of all relative share users. For example, if one user's relative share is 100 and another user's relative share is 200, then the second user gets twice as much access to system

## SET SHARE

resources as the first. If no share is specified in a user's directory entry, then the share defaults to a relative share of 100. The operand *nnnn* ranges from 1 to 10000.

### **INITIAL**

specifies that this user's share of the system resources is to be reset to that specified in the user's directory entry. In the rare case that this user ID no longer appears in the directory, the user's share of the system resources is set to a relative share of 100.

### **ABSolute *mmm%***

specifies that this user is to receive a target maximum of *mmm%* of the scheduled CPU resource; *mmm* is a decimal real number—no or one decimal place—from 0.1 to 100 (for example, 20.5% or 80%).

If ABSOLUTE is not specified, the maximum share has the same type as the minimum share. The *mmm%* must be specified when the minimum is an ABSOLUTE value; *mmmm* must be specified when the minimum is a RELATIVE value.

### **RELative *mmmm***

specifies that this user is to receive a target maximum relative share of *mmmm* of the scheduled CPU resource. The operand *mmmm* ranges from 1 to 10000.

If RELATIVE is not specified, the maximum share has the same type as the minimum share. The *mmm%* must be specified when the minimum is an ABSolute value; *mmmm* must be specified when the minimum is a RELATIVE value.

### **NOLimit**

specifies that a user's share of CPU resource is not limited.

### **LIMITSoft**

specifies that a user's share of CPU resource is limited. However, there are times when LIMITSOFT users will receive more than their limit. This occurs when some users are not using all of their shares (for example, they are waiting for I/O to complete), and there are no NOLIMIT users and no users who have yet to reach their limit and can use additional CPU resource.

If a maximum share value was not specified, the user's minimum share is also its maximum share of the CPU resource.

### **LIMITHard**

specifies that a user's share of CPU resource is limited. When LIMITHARD is specified, a user does not receive more than its maximum share of the CPU resource. If a maximum share is not specified, the minimum share is also the maximum.

If a maximum share value was not specified, the user's minimum share is also its maximum share of the CPU resource.

See the SET SRM LIMITHARD command for details of setting the method that the CP scheduler will use to enforce the limit on the guest's CPU usage.

### **TYPE**

specifies that this change to the user's share setting applies to the specified CPU type and that the share settings for other types are not changed. If TYPE is not specified, or if TYPE ALL is specified, this change applies to all CPU types. The determination of whether this setting has an effect on virtual CPUs depends on the CPU affinity setting for the virtual machine and whether real CPUs of the corresponding type are available in the processor configuration. See Usage Note [“8” on page 1693](#) for further details regarding setting share values by CPU type.

## **Usage Notes**

1. You can set a user's system-resource-access priority for each processor type with the SET SHARE command, but some of these processor types are not available in certain processor configurations. The share values are still displayed for the unavailable processor types, but they are not used.
2. Be aware that a user's share denotes priority access to the entire set of system resources, not just CPU. Currently, the following resources are scheduled according to the share specification:

- CPU
- Real storage
- Paging capacity.

In fact, a bottlenecked resource and a user's requirement for it determine the access priority. For example, if storage is constrained, the user receives access to real storage according to its share. Likewise, if paging is a problem, the user receives paging service according to its share.

- An absolute share user is receiving its correct service when the time it would take to complete a task if it were running alone in the system is the inverse of its share percentage. For example, if a user receives 50% (1/2) of the system, it should take no more than twice as long to complete a unit of work as it would if it were running alone. If it gets 25% (1/4) of the system, it should not take more than four times what it would running alone. The gauge of relative share delivery is slightly different. Two relative share users with similar resource requirements can be compared. If the users are limited by the same bottlenecked resource, a RELATIVE SHARE user with twice the SHARE of another should run twice as fast.
- The target maximum share is applicable only to the CPU resource.
 

If a target maximum share is specified, the user will not necessarily receive that amount of scheduled CPU resource. This is because a user will only receive more than the target minimum share for a resource when other users are not using all of their target minimum shares.
- When the limit option is specified alone (that is, without a share and without a maximum share being specified), only that option is changed. The userid retains its previous share and the maximum share is changed (if necessary) to match the new limit option. For example, if NOLIMIT is specified, then the pre-existing maximum share (if any), is reset to zero. Or, if LIMITHARD or LIMITSOFT is specified and there previously was no maximum share, then the maximum share is set equal to the user's minimum share (with the specified 'hard' or 'soft' characteristic).
- Due to the dynamics of VM systems and scheduling, the maximum share target is not always met. The actual usage should be within five percent of the system. The accuracy may be affected by the following:
  - running guest as a virtual MP
  - guest use of diagnose X'44' or X'9C'
  - running VM second level or on LPAR
  - low system utilization
- When setting SHARE RELATIVE on a userid that has multiple CPUs, the minimum share that the system will assign is one per virtual CPU. For example, if you SET SHARE MAINT RELATIVE 1 and MAINT has five virtual CPUs, the resulting SHARE will be set to five.
- When setting the SHARE of a userid that has multiple CPUs in its virtual configuration, there are special considerations. The share for the virtual machine is divided across all of the virtual processors. In the simplest case, if an absolute share of 10% is set for a virtual machine with two shared virtual CPUs, each gets a 5% share of the system's CPU resources. If one of the processors cannot use its 5% share, the other processor does not get more. Each is entitled to 5% of the system individually. If in the above case one of the virtual CPUs has been stopped and the other is started, then the entire 10% share of system resources would be given to the started virtual CPU.

The user's virtual MP configuration can include more than one type of processor, such as CP, IFL, zIIP, or ICF. In this situation, for scheduling purposes each type of processor is treated as a separate resource. There are two cases depending on whether the virtual machine has CPUAFFINITY ON or CPUAFFINITY OFF. In both cases, assume that all virtual CPUs are started.

If CPUAFFINITY is OFF, there is only one real resource pool on which these virtual CPUs can be dispatched—the primary CPU type. The virtual machine's share is distributed evenly across all of its virtual CPUs regardless of CPU type.

If CPUAFFINITY is ON, the virtual machine's share is distributed evenly across each set of virtual CPUs of the same type. As an example, consider a virtual machine with a share of absolute 24% and with two virtual CPUs—one CP and one zIIP. In this case, each virtual CPU will be given a share of

24% of its corresponding type of real processor time. If there are two real CPs and one real zIIP, the virtual CP will be entitled to a 24% share of each CP or 48% of one real CP. The virtual zIIP is entitled to 24% of the zIIP.

Now consider a virtual machine with a share setting of absolute 24% that has four virtual CPs and two virtual zIIPs, all started. In this example, each virtual CP is given a share of 6% of the real CP resources and each virtual zIIP is given a share of 12% of whatever real zIIP resources exist. However, if there are no real zIIPs, CPUAFFINITY is suppressed for the virtual zIIPs and they are dispatched on the real CPs. In that case, all six virtual CPUs are considered to be sharing the CP resources and each is given a 4% share of the real CP processors.

There are several consequences of all this which must be considered when assigning a share value to a virtual MP userid.

- a. The examples above use absolute shares as they are the easiest to visualize. However, the same concepts hold true for relative share settings.
  - b. Each type of CPU resource must be considered as a separate resource that is controlled by the SHARE setting.
  - c. The CPUAFFINITY setting for a virtual machine might have implications for the total CPU time that this virtual machine can receive. This setting can be changed with the SET CPUAFFINITY command, or by varying on a real processor if it is the first of its type, or by varying off a real processor if it is the last of its type.
  - d. Changing the number of virtual CPUs in a virtual configuration might change the actual share of real CPU resource a virtual machine is entitled to if a new type is added to the virtual configuration or the last of one type is removed from the configuration.
9. CP calculates and enforces consumption limits in a different manner than deadline limits. A virtual machine is consumption limited when the following conditions exist:
- The virtual machine is assigned an ABSOLUTE LIMITHARD share.
  - CONSUMPTION limiting (SET SRM LIMITHARD CONSUMPTION) is in effect on the system.

For consumption limits, CP always assumes that every logical CPU can run to 100% busy, regardless of the LPAR's entitlement and regardless of the availability of excess power on the CPC. For example, if the system has five shared logical IFL CPUs, CP assumes that the shared IFL CPU resource has a total capacity of 500%. If the virtual machine in this case is assigned an ABSOLUTE LIMITHARD share of 50% for TYPE IFL, the virtual machine will be allowed to consume a CPU capacity of 250% (that is, 2.5 CPUs of power). If PR/SM is limiting the LPAR's five logical IFL CPUs to 280% total capacity, CP will still allow the virtual machine to consume 250%. In that case, CP is really allowing the virtual machine to consume  $(250/280) = 89\%$  of the IFL CPU resource available to the LPAR. If DEADLINE limiting (SET SRM LIMITHARD DEADLINE) is in effect on the system, the virtual machine will be allowed to consume approximately  $(50 * 280) = 140\%$  of the IFL CPU capacity, because DEADLINE limiting considers the actual amount of CPU resource available to the LPAR.

10. When APAR VM65680 is applied and multithreading is enabled, prorated core time is used in the consumption limiting calculation for a virtual machine being consumption limited. When multithreading is not enabled or APAR VM65680 is not applied, raw CPU time is used instead. For an explanation of the different measures of CPU time, see [Simultaneous Multithreading \(SMT\) in z/VM: Performance](#).
11. Share settings control access to CPU power on a system-wide, per-processor-type basis, but they have no bearing or influence on how the power associated with a RESPOOL is distributed to its members. When the RESPOOL limit is reached, all members of the pool are put onto the limit list for a while, and then, after a delay sufficient to compensate for any excess use, they are all removed. The scheduler makes no effort to distribute the power of the RESPOOL to its members in accordance with the members' share settings.

## Responses

Response 1:

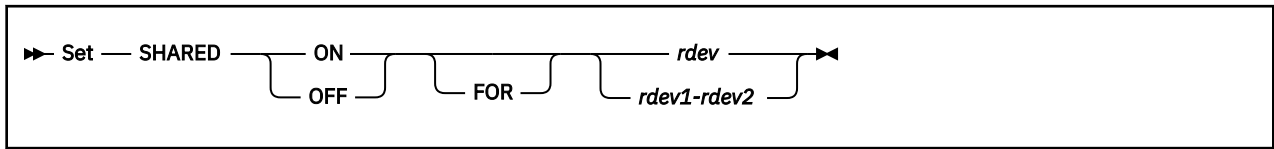
```
USER userid: CP   share_type SHARE = value
                  MAXIMUM SHARE = option share_type value
IFL  share_type SHARE = value
                  MAXIMUM SHARE = option share_type value
ICF  share_type SHARE = value
                  MAXIMUM SHARE = option share_type value
ZIIP share_type SHARE = value
                  MAXIMUM SHARE = option share_type value
```

is issued when the command completes successfully. See [“QUERY SHARE” on page 1161](#) for field descriptions.

## Messages

- HCP007E Invalid *userid* - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on

## SET SHARED



### Authorization

Privilege Class: B

### Purpose

Use SET SHARED to control whether a full-pack minidisk is to be shared by the users of many real and virtual systems.

### Operands

**ON**

**OFF**


indicates that the device is shared or not shared among multiple real and virtual systems.

*rdev*

*rdev1-rdev2*

is the real device number or range of real device numbers of the devices to be shared.

### Usage Notes

1.  **Attention:** Use SET SHARED OFF with extreme caution and only after careful consideration.
2. You should contact the operators of the systems that are sharing the device to assure that they mark a device as unusable in their system configuration and physically disable their interfaces to that device *before* you enter the SET SHARED OFF command.
3. You may use this command to tell CP whether to issue real reserve/release CCWs to the device associated with the full-pack minidisk.
4. When you SET SHARED OFF for a full-pack minidisk, that device is *not* released by CP. If the device is currently reserved by CP for a user, it remains reserved until that user issues a release to that device.
5. You can determine the SHARED status of system DASD by issuing the class B QUERY DASD RESERVE command.
6. When you SET SHARED ON for a device, minidisk caching is turned off for the device. If minidisk caching is desired, it must specifically be set on via the SET MDCACHE command.
7. This command will be rejected for full-pack minidisks if working allegiance (SET WRKALLEG) has been turned on for one of the device's associated minidisks. Working allegiance is simulated to ensure data integrity among users sharing the same minidisk on the same VM system. Sharing real devices on the same minidisk is not allowed.

### Responses

Response 1:

```
SHARED SET {ON } FOR rdev
           {OFF }
```

is issued when the command completes successfully.



## Messages

- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP2003E Shared not set for *type rdev*; not a shared device.
- HCP2004E Shared not set for *type rdev*; does not support reserve/release.
- HCP2006E Shared not set for DASD *rdev*; device not online.
- HCP2007E Shared not set for DASD *rdev*; WRKALLEG is ON for user *userid*, minidisk *vdev*.
- HCP6000E The range of device numbers cannot exceed 256.

## SET SHUTDOWNTIME

```
►► Set — SHUTDOWNtime — interval ◄◄
```

### Authorization

Privilege Class: A, C

### Purpose

Use the SET SHUTDOWNTIME command to specify the CP shutdown interval. The CP shutdown interval is the time that is reserved for CP to shut down when a SHUTDOWN command is issued or the entire z/VM system shuts down automatically as the result of receiving a hardware deactivation signal.

The SET SHUTDOWNTIME command specifies how much of the entire z/VM system-shutdown interval is reserved for CP shutdown. The entire z/VM system-shutdown interval might also include a shutdown-signal timeout interval, which provides time for guest operating systems to shut down. A shutdown-signal timeout interval occurs only if the CP shutdown interval is less than the entire z/VM system-shutdown interval.

### Operands

#### *interval*

specifies the amount of time that is reserved for a CP shutdown to be completed. The value specifies the number of seconds in the range 0 - 32767.

If the CP shutdown time is not specified in the SET SHUTDOWNTIME system configuration statement or by issuing the SET SHUTDOWNTIME command, the default is 30 seconds.

### Usage Notes

1. The CP shutdown interval is an estimation of the time that CP requires to complete an orderly CP shutdown. If CP requires more time to shut down, the actual interval can be longer.
2. The SET SHUTDOWNTIME command dynamically overrides the specified or default value of the SET SHUTDOWNTIME system configuration statement, or the value specified on a previous SET SHUTDOWNTIME command for the current system IPL.
3. The interval that is specified on this command can affect the interval that is designated for the entire z/VM system shutdown process.

If the CP SHUTDOWN command is issued without a WITHIN, BY, or IMMEDIATE operand, the start of the CP shutdown process can be delayed. The start of the CP shutdown process is delayed until the system-default shutdown-signal timeout interval elapses or until all signaled guests indicate that they shut down, whichever occurs first. The time that is designated for the entire z/VM system to shut down is the sum of the system-default shutdown-signal timeout interval and the CP shutdown interval. The entire z/VM system shutdown can take less time than the sum of the system-default shutdown-signal timeout interval and the CP shutdown interval if one or both of the following conditions occur:

- All signal-enabled guests shut down before the system-default shutdown-signal timeout interval elapses.
- The CP shutdown process finishes in less time than is reserved by the CP SET SHUTDOWNTIME specification.

Use caution when you specify the WITHIN, BY, or IMMEDIATE operand on the SHUTDOWN command. The WITHIN, BY, or IMMEDIATE operand specifies the time interval for the entire z/VM system shutdown. The actual shutdown-signal timeout interval is the difference between the entire z/VM

system shutdown interval and the CP shutdown interval. The actual shutdown-signal timeout interval might not provide guests with enough time to complete an orderly shutdown.

For examples of the interaction between the SHUTDOWN command timing operands, the CP shutdown interval, and the system-default shutdown-signal timeout interval, see [“Examples” on page 1794](#) in the SHUTDOWN command topic.

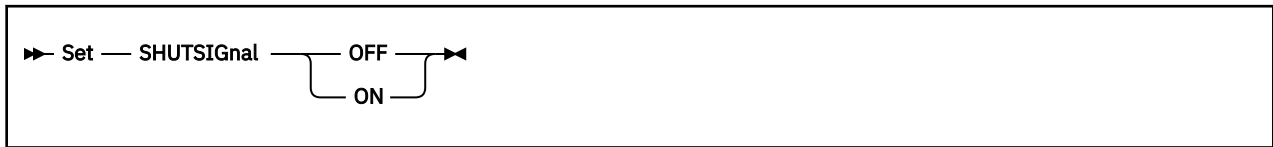
**Tip:**

- The system-default shutdown-signal timeout interval is specified by the SET SIGNAL SHUTDOWNTIME command or system configuration statement. For information about the system-default shutdown-signal timeout interval, see [“SET SIGNAL” on page 1701](#).
  - The CP shutdown interval is specified by the SET SHUTDOWNTIME command or system configuration statement.
4. When a CP shutdown occurs due to a hardware deactivation, if the SHUTDOWNTIME is more than 300 seconds (5 minutes), there is not time for guests to shut down.

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*
- HCP6706E Invalid time - *time*

## SET SHUTSIGNAL



### Authorization

Privilege Class: G

### Purpose

Use SET SHUTSIGNAL to temporarily disable and later to re-enable this user for shutdown signals.

The intended use of SET SHUTSIGNAL OFF is to allow a subset of users to be temporarily disabled from receiving shutdown signals while the remaining set of users that are enabled for shutdown signals are shut down using the SIGNAL SHUTDOWN command. Later, the disabled users can be re-enabled with the SET SHUTSIGNAL ON command, and then signaled to shut down via the SIGNAL SHUTDOWN command. This allows for an ordered shutdown of users.

The SET SHUTSIGNAL command is valid only when a guest operating system is running in the virtual machine and has enabled for shutdown signals.

### Operands

#### OFF

temporarily disables this user from receiving shutdown signals.

#### ON

ends the temporary disablement for shutdown signals imposed by the SET SHUTSIGNAL OFF command.

### Usage Notes

1. SET SHUTSIGNAL OFF should be used with extreme caution because it will prevent disabled users from receiving a shutdown signal when the z/VM system is about to shut down. These users will be stopped with no warning when the z/VM system shuts down.
2. Users that disable for shutdown signals with the SET SHUTSIGNAL OFF command will not be logged off following the timeout of a shutdown signal generated by the SIGNAL SHUTDOWN command.
3. Any shutdown signals that are not presented to a user due to being disabled by the SET SHUTSIGNAL OFF command will not be presented when the user re-enables for shutdown signals with the SET SHUTSIGNAL ON command.
4. Any action initiated by the guest operating system to modify its enablement for shutdown signals will clear any existing disablement for the user from a previously issued SET SHUTSIGNAL OFF command.
5. The QUERY SIGNALS command can be used to determine whether a user is currently enabled to receive shutdown signals.

### Messages

- HCP002E Invalid operand *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP02115E The operating system running in the virtual machine is not enabled for shutdown signals.
- HCP06704E Missing token at end of line

## SET SIGNAL

```
►► Set — SIGNAL — SHUTDOWNtime — interval ◄◄
```

### Authorization

Privilege Class: A, C

### Purpose

Use SET SIGNAL to specify the system-default shutdown-signal timeout interval.

### Operands

#### SHUTDOWNtime *interval*

specifies the duration of the system-default shutdown-signal timeout interval. The *interval* specifies a number of seconds in the range 0 - 32767. If the system-default shutdown-signal timeout interval is not specified by the SET SIGNAL SHUTDOWNTIME command or system configuration statement, the default is 0 seconds.

If a SIGNAL, FORCE, or SHUTDOWN command is issued without specifying an interval, the system-default shutdown-signal timeout interval specifies the interval for guest operating systems to complete their shutdown processing. If the interval value is greater than 0, a shutdown signal is sent to enabled users. If the interval value is 0, shutdown signals are suppressed.

### Usage Notes

1. The SET SIGNAL SHUTDOWNTIME command dynamically overrides the specified or default value of the SET SIGNAL SHUTDOWNTIME system configuration statement, or the value that was specified on a previous SET SIGNAL SHUTDOWNTIME command, for the current system IPL.
2. A user is enabled for signals only when a guest operating system is running in the virtual machine and is enabled to receive these signals.
3. The interval that is specified on this command can affect the interval that is designated for the entire z/VM system shutdown process.

If the CP SHUTDOWN command is issued without a WITHIN, BY, or IMMEDIATE operand, the start of the CP shutdown process can be delayed. The start of the CP shutdown process is delayed until the system-default shutdown-signal timeout interval elapses or until all signaled guests indicate that they shut down, whichever occurs first. The time that is designated for the entire z/VM system to shut down is the sum of the system-default shutdown-signal timeout interval and the CP shutdown interval. The entire z/VM system shutdown can take less time than the sum of the system-default shutdown-signal timeout interval and the CP shutdown interval if one or both of the following conditions occur:

- All signal-enabled guests shut down before the system-default shutdown-signal timeout interval elapses.
- The CP shutdown process finishes in less time than is reserved by the CP SET SHUTDOWNTIME specification.

Use caution when you specify the WITHIN, BY, or IMMEDIATE operand on the SHUTDOWN command. The WITHIN, BY, or IMMEDIATE operand specifies the time interval for the entire z/VM system shutdown. The actual shutdown-signal timeout interval is the difference between the entire z/VM system shutdown interval and the CP shutdown interval. The actual shutdown-signal timeout interval might not provide guests with enough time to complete an orderly shutdown.

## SET SIGNAL

For examples of the interaction between the SHUTDOWN command timing operands, the CP shutdown interval, and the system-default shutdown-signal timeout interval, see [“Examples” on page 1794](#) in the SHUTDOWN command topic.

### Tip:

- The system-default shutdown-signal timeout interval is specified by the SET SIGNAL SHUTDOWNTIME command or system configuration statement.
- The CP shutdown interval is specified by the SET SHUTDOWNTIME command or system configuration statement. For information about the CP shutdown interval, see [“SET SHUTDOWNTIME” on page 1698](#).

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*
- HCP6706E Invalid time - *time*

## SET SMSG



### Authorization

Privilege Class: G

### Purpose

Use SET SMSG to control whether your virtual machine accepts special messages.

### Operands

#### ON

#### OFF

determines whether your virtual machine receives special messages. If you specify OFF, the virtual machine cannot receive special messages. If you specify ON, the virtual machine can receive special messages. At logon, SMSG is set to OFF.

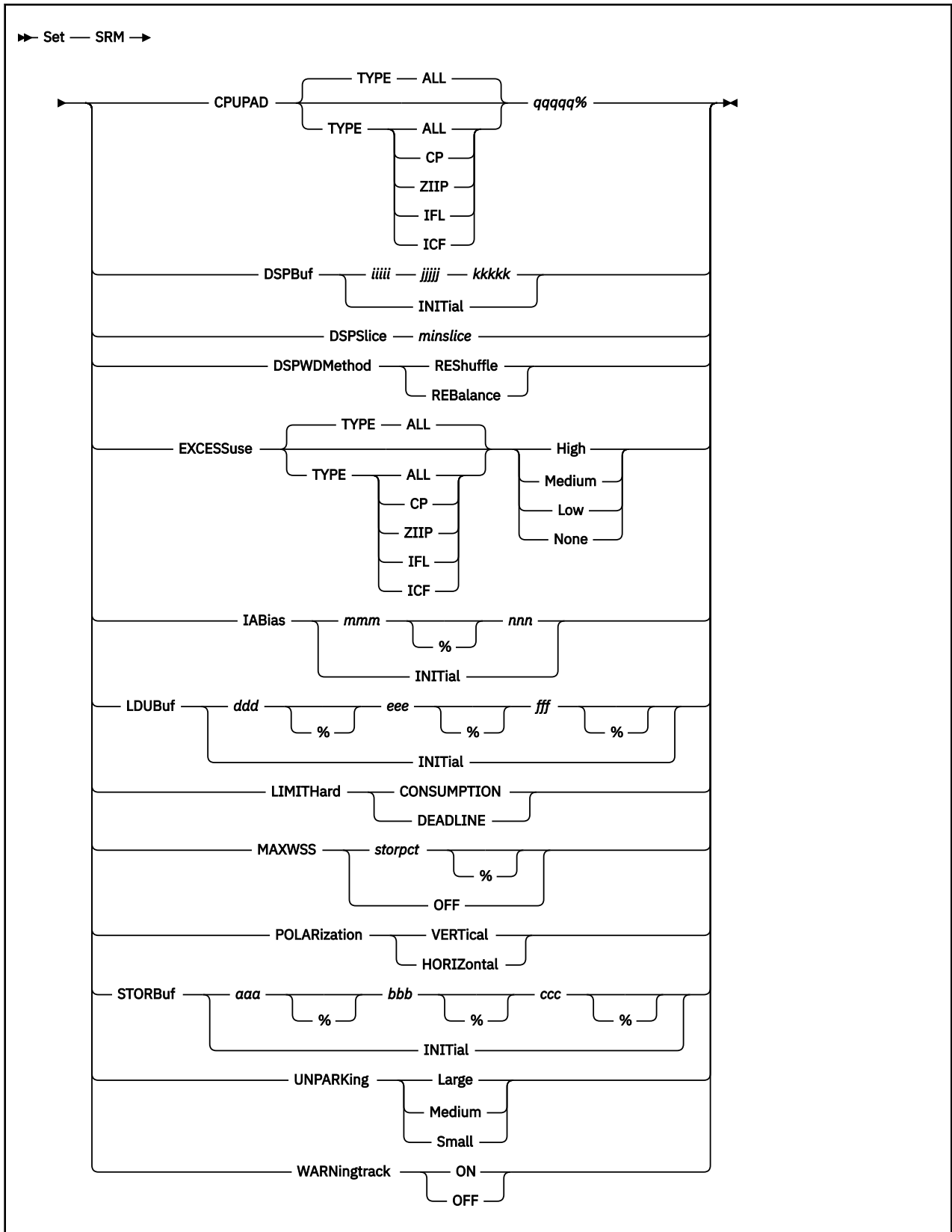
#### IUCV

specifies that special messages are to be passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the message is handled as if SET SMSG ON had been issued.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid

# SET SRM





## Authorization

Privilege Class: A

## Purpose

Use SET SRM (system resource manager) to change system parameters.

## Operands

### CPUPAD TYPE

specifies the CPU type pool to which the CPUPAD setting applies.

### CPUPAD *qqqqq%*

specifies the amount of additional CPU capacity the system should be prepared to consume beyond what the system projects it will need. *qqqqq* is a value from 0 through 08000, which represents the amount of additional CPU capacity as a percentage of an entire CPU. A value of 100 represents an entire CPU. This setting applies only when the system is running in vertical polarization mode. When the system's projection plus the specified amount of additional CPU capacity exceeds the partition's entitlement, this value might not be used in determining how many CPUs are unparked. A value of 100 is the system default setting.

For additional information on CPUPAD and HiperDispatch, see the [SRM Statement](#) and [The Objective of z/VM HiperDispatch](#) in *z/VM: CP Planning and Administration*.

When multithreading is enabled, this value specifies the amount of additional core capacity the system should be prepared to consume beyond what the system projects it will need. A value of 100 represents an entire core.

### DSPBuf *{iiii jjjj kkkkk}*

gives the scheduler criteria for admitting users to the dispatch list in cases of users who are not prevented from entering by the more resource-specific barriers established by STORBUF and LDUBUF. DSPBUF stands for “dispatch buffer” and refers to the total number of users (by class) that are allowed in the dispatch list.

**Note:** CP's virtual processor management has been improved so that no user stays in the eligible list more than an instant before being added to the dispatch list. Therefore some functions intended to improve performance by managing the eligible list, such as the DSPBUF option, are now less meaningful.

The operand *iiii* refers to the number of openings in the dispatch list made available to users with short-running (E1) transactions. A subset, *jjjj*, of the *iiii* openings is also made available to users with medium-length transactions; the E1 and E2 users compete for these slots. Another subset, *kkkkk*, of the *jjjj* openings is also made available to users with long-running transactions; the E1, E2, and E3 users all compete for these slots. QUICKDSP and other E0 users are not limited by these numbers and may enter the dispatch list without limit.

The valid range of values is 1 to 32767 where  $iiii \geq jjjj \geq kkkkk$

**Example:** `dsdbuf 35 30 18`

35 openings are available to E1 users, but only 5 of them ( $5 = 35 - 30$ ) are exclusively available to E1 and are guaranteed not to be already occupied by E2 and E3 users.

30 openings are available to E2 users, but only 12 of them ( $12 = 30 - 18$ ) are guaranteed not to be already occupied by E3 users. None of the 30 openings is guaranteed against occupation by E1 users; however, E1 users generally get out of the way quickly.

18 openings are available to E3 users, but none of the openings is guaranteed against occupation by E1 and E2 users. When E1 and E2 users do occupy these openings, you can expect them to get out of the way relatively quickly compared with the time scale of an E3 transaction.

**DSPBuf INITIAL**

sets the parameters back to their initial values which, for practical purposes, are infinity.

**DSPSlice minslice**

is the new size, in milliseconds, of the dispatching minor time slice. This is the amount of CPU time a virtual machine may consume before reexamination by the scheduler. The value *minslice* must be in the range of 1 to 100.

**DSPWDMethod REShuffle**

sets the dispatcher work distribution algorithm to reshuffle. This algorithm periodically alters the home assignments of active virtual CPUs in an attempt to balance the number of virtual CPUs on all dispatch vectors. Unless a system configuration file SRM statement or a CP SET SRM command changes the work distribution algorithm, the system uses the reshuffle algorithm.

When multithreading is enabled, the only algorithm supported is reshuffle.

**DSPWDMethod REBalance**

sets the dispatcher work distribution algorithm to rebalance. This algorithm tracks virtual machine CPU utilization and periodically alters the home assignments of virtual CPUs in an attempt to balance the CPU demand on dispatch vectors. This option should be used with caution. Use this option only when specifically instructed to do so by IBM.

For additional information on DSPWDMETHOD and HiperDispatch, see the [SRM Statement and The Objective of z/VM HiperDispatch in z/VM: CP Planning and Administration](#).

When multithreading is enabled, the only algorithm supported is reshuffle.

**EXCESSuse**

specifies how aggressively the system should attempt to use unentitled CPU capacity. This setting applies only when the system is running in vertical polarization mode, and affects how aggressively vertical-low CPUs are used to consume unentitled CPU capacity.

**EXCESSuse TYPE**

specifies the CPU type pool to which the EXCESSUSE setting applies.

**EXCESSuse High**

specifies that the system should be aggressive in using unentitled CPU capacity.

**EXCESSuse Medium**

specifies that the system should be moderately aggressive in using unentitled CPU capacity. This is the system default setting.

**EXCESSuse Low**

specifies that the system should not be aggressive in using unentitled CPU capacity.

For additional information on EXCESSUSE and HiperDispatch, see the [SRM Statement and The Objective of z/VM HiperDispatch in z/VM: CP Planning and Administration](#)

**EXCESSuse None**

specifies that vertical-low logical cores should always be parked when global performance data (GPD) is available (GPD-on).

**IABias [mmm[% ] nnn]**

specifies the new "interactive bias" to take effect. An interactive bias with a nonzero *mmm* causes users doing trivial transactions to receive better service than their scheduling share would otherwise entitle them to receive. Interactive bias causes a user's share of the system to be momentarily larger than normal at the start of a new transaction. As the transaction progresses, the user's share fades back to normal. This means that transactions requiring only a small fraction of processing time get good response time.

The value *mmm* is a percentage of the distance from the user's normal dispatch-list position to the topmost user. It controls the "intensity" of the interactive bias. This value is a decimal number from 0 to 100, with initial IPL value of 90. The intensity determines where in the dispatch list the user is inserted for the first minor time slice at the start of a new transaction. The insertion point is somewhere between the normal (unbiased) insertion point and the topmost user in the list. A value of 100 gives a very strong interactive bias, and 0 eliminates the interactive bias.

The value *nnn* controls the duration of the interactive bias. This value is a decimal number from 1 to 100, with initial IPL value of 2. The duration specifies the number of minor time slices it takes for users to fade back to their normal position in the dispatch list after the initial interactive boost. For example, a setting of 3 means the boost to the user lasts for 3 minor time slices. The effect is strongest on the first time slice and fades away until on the third time slice the user receives only a small boost in the dispatch list. On the fourth and subsequent time slices the user receives no interactive boost at all.

If *mmm* is specified without *nnn*, the current interactive bias duration is not changed.

#### **IABias INITIAL**

sets the intensity (*mmm*) and duration (*nnn*) back to the initial values, 90 and 2.

#### **LDUBuf {ddd[%] eee[%] fff[%]}**

partitions the commitment of the system's paging resources when the scheduler chooses users to move from the eligible list to the dispatch list. The result is a paging capacity buffering mechanism based on the transaction class of a user. (See the description of the STORBUF parameter in this command for a definition of transaction classes.)

**Note:** CP's virtual processor management has been improved so that no user stays in the eligible list more than an instant before being added to the dispatch list. Therefore some functions intended to improve performance by managing the eligible list, such as the LDUBUF option, are now less meaningful.

During system initialization, the number of paging DASD exposures is determined. Users expected to have high paging rates in the dispatch list are termed "loading users." They are identified by recent DASD paging activity or with the knowledge that their working sets have been moved to DASD since they last ran. One loading user is regarded as keeping one paging exposure busy. The LDUBUF (loading user buffer) percentages, then, determine the number of loading users allowed in the dispatch list by transaction class.

The SET SRM LDUBUF parameters partition the paging capacity as follows:

#### **ddd[%]**

is the percentage of paging exposures the scheduler is to view when considering adding a loading E1, E2, or E3 user to the dispatch list. If the total number of loading users currently in the dispatch list plus the loading *En* user under consideration exceeds the *ddd* percentage of paging exposures, the user may not be added immediately and is delayed further.

#### **eee[%]**

is the percentage of paging exposures the scheduler is to view when considering adding a loading E2 or E3 user to the dispatch list. If the number of Q2 and Q3 loading users currently in the dispatch list plus the loading E2 or E3 user under consideration exceeds the *eee* percentage of paging exposures, the user may not be added immediately and is delayed further.

#### **fff[%]**

is the percentage of paging exposures the scheduler should view when considering adding a loading E3 user to the dispatch list. If the number of Q3 loading users currently in the dispatch list plus the loading E3 user under consideration exceeds the *fff* percentage of paging exposures, the user may not be added immediately and is delayed further.

Note that in order to enter the dispatch list, an E1 user must only pass the one barrier established by *ddd*. But an E2 user must pass the two barriers set by *ddd* and *eee*. And an E3 user must pass the three barriers set by *ddd*, *eee*, and *fff*. This is exactly analogous to the operation of STORBUF, though STORBUF was explained differently. Similarly, the differences between the values (*ddd-eee* and *eee-fff*) represent LDUBUF buffers (for Q1 and Q2 respectively), just as they do for STORBUF.

In each case, the LDUBUF values must be in the range 0 through 999, with a restriction that *ddd* ≥ *eee* ≥ *fff*. Any value greater than 100 represents an overcommitment of DASD paging capacity.

#### **LDUBuf INITIAL**

sets the percentage parameters back to their initial values, 100%, 75%, and 60%.

**LIMITHard**

sets the enforcement of hard limiting of scheduled system resources. This setting only affects users with absolute maximum shares defined with the SET SHARE command or the SHARE directory statement.

**LIMITHard CONSUMPTION**

is the method for enforcing hard limiting based on the guest's measured processor utilization. The CONSUMPTION method is generally more accurate than the DEADLINE method. CONSUMPTION is the system default that is set at initialization time and remains in effect unless the SET SRM LIMITHARD command is issued to change it.

**LIMITHard DEADLINE**

is the traditional method for enforcing hard limiting. It enforces limits based on a calculated deadline and defines the projected instant in time when a guest should complete a timeslice. In most workloads, the CONSUMPTION method will give more accurate results.

**MAXWSS storpct[%]**

sets the maximum working set that any normal user is allowed to have. It is specified as a percentage of the system's pageable real storage. (Pageable real storage is also known as the *dynamic paging area*, *DPA*, or *paging pool*.) If a user's working set size (WSS) grows and exceeds this specified percentage of the pageable real storage, the user is dropped from the dispatch list and put back in the eligible list.

When a user is put in the eligible list for exceeding the MAXWSS, that user is treated as though the user's stay in the dispatch list had run to completion. The user is advanced to the next user class (for example, Q3) and is placed well out in the eligible list so that the user does not immediately get back into the dispatch list to run again. The intent is to delay the user for a while to prevent the user from occupying any more pages than the user already does: the installation has told the scheduler (through the MAXWSS setting) that a user requires a working set size too large to be run. (For example, the installation does not want such large WSS users competing with the normal, more interactive, workload.) Still, the user travels through the eligible list and gets another chance to run, although it is likely that all of this user's page frames will, in the meantime, have been stolen and given to other users. When the user does get to run again, the scheduler again evaluates the working set size: a user with too great a working set size (WSS) requirement gets to run only briefly before the MAXWSS mechanism removes the user from the dispatch list again.

For all this to work as intended, the system must have an actual eligible list—that is, the scheduler must consider the system to be storage constrained according to the setting of the SET SRM STORBUF parameters.

If the MAXWSS percentage is set larger than the STORBUF percentages, the STORBUF percentages take precedence. A user is dropped from the dispatch list and sent to the eligible list if the user's WSS grows so large that it exceeds the STORBUF percentage for that user's class. When this happens, the user never gets a chance to grow large enough to reach the MAXWSS percentage.

Effectively, the MAXWSS control is inoperative when it is set larger than the STORBUF percentages. The value specified for MAXWSS may range up to 9999%.

Users having absolute shares are given special consideration. If a user's absolute share is larger than the MAXWSS percentage, the user is allowed to grow beyond the MAXWSS percentage and to become as large as the user's absolute share percentage. However, such users are still limited by the STORBUF percentage. Having a large absolute share allows a user to exceed the MAXWSS control but not the STORBUF control.

The MAXWSS control does not affect Q0 users, such as users with the QUICKDSP attribute.

**MAXWSS OFF**

sets the MAXWSS percentage to 9999%, effectively turning the mechanism off. (In other words, with the limit set high enough, it is in effect no longer a limit.) This is the initial setting.

**POLARization VERTical**

requests that z/VM and PR/SM operate the partition in vertical polarization mode. In vertical polarization mode, the partition's logical CPUs have differing entitlements to physical CPU resources. Some logical CPUs are entitled to an entire physical CPU, while others are entitled to only a fraction of

a physical CPU or perhaps to only the unused CPU entitlements left unconsumed by other partitions. Further, in a vertical polarization mode partition, PR/SM dispatches high entitlement logical CPUs on the same physical CPUs over and over again, thereby preserving accumulated cache contents. When z/VM runs in a vertical polarization mode partition, z/VM tends to run the system's workload on fewer logical CPUs, tends to keep MP guests' virtual CPUs together with respect to the partition's cache topology, and tends not to move virtual CPUs among logical CPUs. In aggregate, these behaviors tend to improve system performance.

When multithreading is enabled, only vertical polarization is supported.

### **POLARization HORIZontal**

requests that z/VM and PR/SM operate the partition in horizontal polarization mode. In a horizontal polarization mode partition, each online (configured) logical CPU of a particular CPU type gets an equal portion of the partition's weight for that CPU type. For instance, if a partition with 16 online logical CPUs of type CP had a PR/SM weight which entitled those logical CPUs to 8 real CPUs of type CP, then each of the 16 logical CPUs would be entitled to 50% of a real CPU; and when all other partitions consume their entitlement, these 16 will get no more than their 50% of a real CPU. Further, in a horizontal polarization mode partition, PR/SM puts less emphasis on holding logical CPUs fixed on physical CPUs, and z/VM tends to spread work evenly over the online logical CPUs and tends to let virtual CPUs move from one logical CPU to another fairly freely.

When multithreading is enabled, only vertical polarization is supported.

For additional information on POLARIZATION and HiperDispatch, see the [SRM Statement in z/VM: CP Planning and Administration](#).

### **STORBuf {aaa[%] bbb[%] ccc[%]}**

partitions the scheduler's view of real storage when it chooses users to move from the eligible list to the dispatch list. The result is a buffering mechanism based on the transaction class of a user:

#### **E1, Q1**

– users with short-running transactions

#### **E2, Q2**

– users with medium-running transactions

#### **E3, Q3**

– users with long-running transactions,

$E_n$  is the class assigned to a user in the eligible list, and  $Q_n$  is the designation after moving to the dispatch list. An E1 user in the eligible list becomes a Q1 user in the dispatch list, for example. Also, there is transaction class Q0, which is users who never wait in the eligible list to contend for resources in the dispatch list (for example, users with the QUICKDSP attribute).

**Note:** CP's virtual processor management has been improved so that no user stays in the eligible list more than an instant before being added to the dispatch list. Therefore some functions intended to improve performance by managing the eligible list, such as the STORBUF option, are now less meaningful.

The STORBUF values partition pageable storage. They are (in reverse order, for clarity):

#### **ccc[%]**

is the maximum percentage of storage that dispatch list Q3 users can occupy with their working sets.

#### **bbb[%]**

is the maximum percentage of storage that Q2+Q3 users together can occupy. The percentage of storage represented by the difference  $bbb-ccc$  represents a buffer of storage available to Q2 users that the Q3 users are not allowed to use. This buffer protects the Q2 users from delays that occur when slow-moving Q3 users take more (or all) of storage.

#### **aaa[%]**

is the maximum percentage of storage that Q1+Q2+Q3 users can occupy. The percentage of storage resulting from the difference  $aaa-bbb$  represents a buffer of storage available to Q1 users

that the Q2 and Q3 users are not allowed to use. Without this buffer Q2 and Q3 users might occupy all of storage, causing long delays to E1 users who could not enter the dispatch list until one of the slow-moving Q2 or Q3 users left. With the buffer, E1 users enter the dispatch list fairly quickly. Since the buffer is never occupied by Q2 or Q3 users, an E1 user can either enter the dispatch list immediately (if the buffer is not filled), or at worst have only to wait for some Q1 user (in the buffer) to leave.

In each case, the STORBUF values must be decimal numbers from 0 through 999, and  $aaa \geq bbb \geq ccc$ . Any value over 100 is an overcommitment of storage.

Q0 users are left out of the above discussion to keep it simple. In fact, Q0 users are allowed to crowd into storage without limit. In effect then, all of storage above the  $aaa$  percentage can be regarded as a buffer reserved for Q0 users. (Q0 users should not take too much of it, because this unspecified buffer also serves to hold the pages of active users who are temporarily “dormant” [between transactions].)

Although Q3 users may occupy the percentage of storage represented by  $ccc$ , they do not have exclusive rights to it. This storage can be occupied by Q0, Q1, Q2, and Q3 users, depending on which classes are generating a demand for it. But the Q1 and Q2 buffers tend to eliminate some of the Q1 and Q2 demand, so the  $ccc$  storage tends to be occupied mainly by Q3 users. Also, after a Q3 user gets in, the user stays there for a relatively long time, and this gives the Q3 users something of an advantage over other users when it comes to  $ccc$  storage.

Similarly the Q2 users do not have exclusive rights to the buffer represented by the difference  $bbb-ccc$ . This storage can be occupied by Q0, Q1, and Q2, depending on which classes are generating a demand for it. But for the most part it is occupied by Q2 users, just as the  $ccc$  storage is occupied mostly by Q3 users.

The Q1 users do not have exclusive rights to the buffer represented by the difference  $aaa-bbb$ . Both Q0 and Q1 users may occupy it.

#### **STORBuf INITIAL**

sets the percentage parameters back to their initial values, 300%, 250%, and 200%.

#### **UNPARKing**

specifies the unparking heuristic that is to be used when global performance data (GPD) is available.

#### **UNPARKing Large**

unparks all of the logical cores that are powered, in other words, all of the entitled cores and some of the vertical-low cores.

#### **UNPARKing Medium**

unparks all of the vertical-high logical cores, all of the vertical-medium logical cores, and whichever is fewer of the following:

1. all of the vertical-low logical cores that z/VM predicts PR/SM will power
2. all of the vertical-low logical cores that z/VM predicts will be needed to help cover the workload plus the CPUPAD setting.

#### **UNPARKing Small**

unparks only the logical cores that are needed to cover the workload plus the CPUPAD setting.

#### **WARNingtrack**

enables or disables exploitation of the warning-track-interruption facility by CP. This facility notifies CP when PR/SM is about to stop running a logical processor on a physical processor. CP can then save the context of the work being dispatched and yield the logical processor voluntarily, so that the interrupted work is eligible to run on another logical processor. Enabling warning track can improve guest response time and overall performance of workloads that are run on vertical-low or vertical-medium logical processors. When the associated machine facility is available, warning track is turned on by default.

#### **WARNingtrack ON**

enables exploitation of the warning-track-interruption facility by CP.

#### **WARNingtrack OFF**

disables exploitation of the warning-track-interruption facility by CP.

## Usage Notes

1. In using SET SRM DSPSLICE, the initial value of the dispatching minor time slice is determined by the model of the real machine that z/VM is IPLed on. Making the time slice smaller increases system overhead, but it also gives better sorting of the dispatch list and more accurate share control by the scheduler. Making the time slice larger decreases system overhead at the expense of the accuracy of the scheduler share controls. Making the time slice larger may also interfere with the scheduler's ability to quickly sort the dispatch list to get the most I/O-bound users near the top. This in turn may reduce the system's ability to drive I/O and maintain performance.

Remember that when the dispatch slice is changed, the IABIAS duration no longer represents the same amount of processing as before. If the size of the DSPLICE is changed significantly, you may want to adjust the IABIAS duration to compensate so that it still represents roughly the same amount of processing as before.

2. An installation can use the SET SRM STORBUF command to undercommit or overcommit storage. Note that Q0 users are not restricted by their storage requirement. By setting the first percentage value (*aaa*) at less than 100%, you can reserve a storage buffer for Q0 users. It is also sometimes desirable to do this to reserve a storage buffer for active users who are temporarily “dormant” (between transactions). When *aaa* is set to less than 100%, the storage above *aaa* is protected from the (non-Q0) users who are currently running. This means the running users are prevented from crowding all the pages of dormant users out of storage. This is useful when many dormant users' pages are needed sooner than the system can page them out and then in again.

By increasing or decreasing the differences in the three percentage values of STORBUF, an installation can affect the response times and expansion factors for short-running (generally interactive), medium-running, and long-running work.

3. SET SRM LDUBUF is similar to SET SRM STORBUF, except that the resource buffered is the paging configuration capacity, rather than storage. An installation may use SET SRM LDUBUF to undercommit or overcommit the access to paging, or to favor one class of transaction over another.

The values set for SET SRM LDUBUF are usually not critical. However, in most situations, the default values are too low and a better choice would be:

```
SET SRM LDUBUF 300% 200% 100%
```

This represents an overcommitment. If there is a need to do a lot of paging, an overcommitment insures that enough loading users are allowed in the dispatch list to keep the DASD paging devices running at or near their full capacity.

4. z/VM does not virtualize system topology information for its guests. Thus, when z/VM runs in a virtual machine (second level), it runs in horizontal polarization mode and the SET SRM POLARIZATION VERTICAL command produces an error message.
5. In using SET SRM EXCESSUSE:
  - The EXCESSUSE setting affects GPD-on unparking only.
  - SET SRM EXCESSUSE selects the vertical-low policy for all core types.
  - SET SRM EXCESSUSE TYPE selects the vertical-low policy for the specified core type: ALL, CP, ICF, IFL, or ZIIP.
  - When EXCESSUSE NONE is in effect for a core type and GPD is on, z/VM always parks all vertical-low logical cores of that type.
  - Invalid syntax produces message HCP026E or message HCP002E, as appropriate.
6. In using SET SRM UNPARKING:
  - The unparking heuristic can be set at any time, but the heuristic is in effect only when the LPAR is vertical and GPD is on.
  - It might take up to two seconds for the selected heuristic to take effect because z/VM makes unparking decisions only every two seconds.
  - Specifying something other than Large, Medium, or Small produces message HCP002E.

## 7. In using SET SRM WARNINGTRACK:

- z/VM does not virtualize the warning-track-interruption facility for its guests. Thus, when z/VM runs in a virtual machine (second-level), it runs with warning-track exploitation off. This cannot be changed.
- When the warning-track-interruption facility is available and the warning-track setting has not been specified using the SRM WARNINGTRACK configuration statement, the default setting is ON.
- Setting SRM WARNINGTRACK to ON is most helpful in vertically-polarized configurations that include vertical-low and vertical-medium logical processors.

**Responses**

## Response 1:

```
DSPBUF : Q1=iiii Q2=jjjj Q3=kkkk
```

is returned for the SET SRM DSPBUF command.

## Response 2:

```
DISPATCHING MINOR TIMESLICE = xxx MS
```

is returned for the SET SRM DSPSLICE command.

## Response 3:

```
IABIAS : INTENSITY=nnn; DURATION=mmm
```

is returned for the SET SRM IABIAS command.

## Response 4:

```
LDUBUF : Q1=ddd% Q2=eee% Q3=fff%
```

is returned for the SET SRM LDUBUF command.

## Response 5:

```
LIMITHARD METHOD: method
```

is returned for the SET SRM LIMITHARD command.

## Response 6:

```
MAXWSS : LIMIT=nnnn%
..... : PAGES=mmmmmmmmmm
```

is returned for the SET SRM MAXWSS command. The two-line response shows the new MAXWSS value as both a percentage of pageable real storage and the corresponding number of page frames.

## Response 7:

```
STORBUF : Q1=aaa% Q2=bbb% Q3=ccc%
```

is returned for the SET SRM STORBUF command.

## Response 8:

```
POLARIZATION: type
GLOBAL PERFORMANCE DATA: ON|OFF|NA
```

is returned for the SET SRM POLARIZATION command.

## Response 9:

```
EXCESSUSE: CP-excess IFL-excess ICF-excess ZIIP-excess
```



is returned for the SET SRM EXCESSUSE command. The response shows the EXCESSUSE setting for each CPU type.

Response 10:

```
CPUPAD: CP-qqqqq% IFL-qqqqq% ICF-qqqqq% ZIIP-qqqqq%
```

is returned for the SET SRM CPUPAD command. The response shows the CPUPAD setting for each CPU type.

Response 11:

```
DSPWMETHOD: RESHUFFLE|REBALANCE
```

is returned for the SET SRM DSPWMETHOD command.

Response 12:

```
UNPARKING: { LARGE | MEDIUM | SMALL }
```

is returned for the SET SRM UNPARKING command.

Response 13:

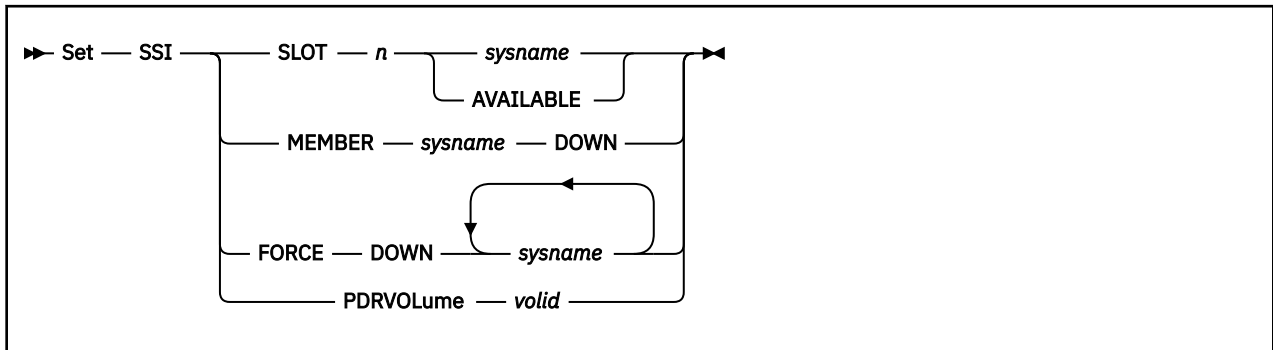
```
WARNING TRACK: AVAILABLE|UNAVAILABLE ON|OFF
```

is returned for the SET SRM WARNINGTRACK command.

## Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP1050E SET SRM POLARIZATION not performed; CPU topology is not available.
- HCP1050E SET SRM POLARIZATION not performed; multithreading is enabled.
- HCP1050E SET SRM DSPWMETHOD not performed; multithreading is enabled.
- HCP1050E SET SRM WARNINGTRACK not performed; warning-track-interruption facility is not available.
- HCP1052I Global performance data is disabled. This may degrade system performance.
- HCP1054E SET SRM LIMITHARD DEADLINE cannot be issued while a resource pool exists
- HCP1401I Expanded storage is not supported.

## SET SSI



### Authorization

Privilege Class: A

### Purpose

Use SET SSI to change existing entries in the single system image (SSI) member list, to notify the SSI cluster that a non-communicating member is down, or to change the volume that holds the Persistent Data Record (PDR). The SET SSI command affects all members of the SSI cluster and is valid only if your system is in an SSI cluster. This command allows you to modify the SSI configuration that was defined by the SSI statement in the system configuration file. For more information about the [SSI Statement](#), see *z/VM: CP Planning and Administration*.

### Operands

#### SLOT *n*

specifies the number of the slot in the SSI member list that you are changing. The number *n* cannot exceed the number of slots currently defined in the SSI member list. You can use the QUERY SSI command to find the number of slots that are currently available, though all available slots might not be usable at this time. For more information, see usage note “11” on page 1716.

#### *sysname*

specifies the system name to add to the SSI member list. You cannot specify the same system name in more than one slot in the SSI member list.

#### AVAILABLE

tells CP to disassociate the slot from a specific member and make it available for allocation.

#### MEMBER *sysname* DOWN

notifies the SSI cluster that the specified member is down to enable the SSI cluster to be returned to a stable mode. The member state must be unknown and the member should be inactive in order to be declared down. The SET SSI MEMBER *sysname* DOWN command is not allowed when that member is active. A member is determined to be active if it appears to be responsive. The operator is responsible for verifying that the member is, and will, remain down.

#### FORCE DOWN *sysname*

notifies the SSI cluster that the specified members are down in order to enable the SSI cluster to be returned to a stable mode. The state of all specified members and their heartbeat activity are not checked. The operator is responsible for verifying that the specified members are, and will, remain down.

This operand is intended to be used by automated operator virtual machines that have external knowledge about the status of the members of the SSI cluster. This operand should not be used by normal system operations. Specifying members that are not truly down will cause data corruption.

This operand is only allowed if the local system is Joined or Suspended. It is not allowed if the local system is in Leaving, Isolated, or Joining state.

### **PDRVOLUME *valid***

causes the DASD volume with the specified label to become the current Persistent Data Record (PDR) device. The SET SSI PDRVOLUME command will be accepted only when the SSI cluster is in the stable mode and all joined members support this PDRVOLUME option. The new volume must be online and attached to the system on all joined members and must contain a PDR specifying the SSI cluster name. The current PDR is copied to the new device and the PDR on the old device is marked obsolete.

### **Usage Notes**

1. Unless the system configuration file has been updated, all changes applied by the SET SSI command will result in a configuration mismatch at the next system initialization. However, in the case of a successful PDR relocation by a SET SSI PDRVOLUME command, a member system attempting to join the SSI cluster after the new PDR volume is activated but before the configuration file has been updated will use the new PDR automatically. For other cases, this will most likely result in an error message and a system wait state. Always update the system configuration file promptly after each successful SET SSI command. After a successful SET SSI PDRVOLUME command, you should update the SSI configuration statement with the replacement PDR volume ID and you must ensure that the system configuration file lists the new volume in either the CP-owned list or the user volume list on all SSI members.
2. All changes applied by SET SSI command are made to all members that are joined to the SSI cluster.
3. The SET SSI command is not allowed on members that are in Isolated state or when the SSI cluster is in Influx or Safe mode with two exceptions:
  - a. If the SSI cluster is in Safe mode due to a member being in Unknown state, the SET SSI *sysname* DOWN command can be entered to mark the member that is in Unknown state DOWN.
  - b. If the SSI cluster is in Safe or Influx mode, the SET SSI FORCE DOWN command can be entered to mark any member DOWN except for the local system.
4. The SET SSI command is not allowed on a system which has been IPLed using the REPAIR IPL parameter.
5. If the local system is not a member of the SSI cluster, the SSI system configuration file statement must be used to define the local system in the SSI member list.
6. Use the QUERY SSI command to display the SSI member list, the number of available slots, and information about the current PDR volume. For more information, see [“QUERY SSI” on page 1177](#).
7. Slot definitions may be changed from AVAILABLE to a member name, or from a member name to AVAILABLE. Changes from one member name to another are not allowed.
8. A slot can not be changed to AVAILABLE status unless the member associated with the slot has been shutdown.
9. When a SNAPDUMP command is issued on a Joined member of an SSI cluster, the SNAPDUMP processing makes the member unresponsive to other Joined members in the SSI cluster. Therefore, it is possible for the SNAPDUMP processing to cause the other Joined members of the SSI cluster to become Suspended, enter SAFE mode, and change the state of the member taking the SNAPDUMP to Unknown.

If a SET SSI MEMBER *sysname* DOWN

or SET SSI FORCE DOWN command is then issued to mark the system taking the SNAPDUMP Down, the system that was marked Down will continue running once the SNAPDUMP is complete. At that point, the system will recognize that it was marked Down in the PDR and take a PLM018 abend (because data corruption could occur from a Joined member running when it is marked Down in the PDR).

10. After the successful completion of a PDR relocation to a new device, the PDR on the old device is marked obsolete. To allow an old PDR volume to be used again, initialize it using the FORMSSI utility.

Alternatively, you can switch back to the previous PDR volume with another SET SSI PDRVOLUME command if the PDR marked obsolete remains intact and still names the same SSI cluster.

- Slots 5-8 cannot be set to anything other than AVAILABLE until all joined members of the SSI cluster are using the 8-member spool ID assignment algorithm, as set on the SPOOL\_MEMBERS operand of the SSI\_CONTROLS statement in the system configuration file. To determine which members are not yet enabled, use the QUERY SSI SSI\_CONTROLS command.

If the output of the QUERY SSI command contains only four members, 8-member support is not installed on this member, so only slots 1-4 can be defined.

If the output of the QUERY SSI SSI\_CONTROLS command indicates that any joined members are using the 4-member spool ID assignment algorithm, only slots 1-4 can be defined. Slots 5-8 are displayed as AVAILABLE but cannot be defined until all joined members are using the 8-member algorithm.

If the output of the QUERY SSI SSI\_CONTROLS command indicates that all joined members are using the 8-member spool ID assignment algorithm, all 8 slots can be defined.

## Examples

- Querying the current SSI cluster displays the following information:

```
query ssi
SSI Name:  ALEAST
SSI Mode:  Stable
SSI Persistent Data Record (PDR) device: SSIVOL on 3333
SLOT SYSTEMID STATE      PDR HEARTBEAT      RECEIVED HEARTBEAT
  1 YANKEES  Joined    01/30/2010 12:03:33 01/30/2010 12:03:35
  2 REDSOX   Joined    01/30/2010 12:03:47 01/30/2010 12:03:49
  3 ----- Available
  4 ----- Available
  5 ----- Available
  6 ----- Available
  7 ----- Available
  8 ----- Available
```

To add system ORIOLES to the SSI member list in the available slot 3, enter:

```
set ssi slot 3 orioles
```

Subsequently, querying the updated SSI cluster displays the following information:

```
query ssi
SSI Name:  ALEAST
SSI Mode:  Stable
SSI Persistent Data Record (PDR) device: SSIVOL on 3333
SLOT SYSTEMID STATE      PDR HEARTBEAT      RECEIVED HEARTBEAT
  1 YANKEES  Joined    01/30/2010 12:03:33 01/30/2010 12:03:35
  2 REDSOX   Joined    01/30/2010 12:03:47 01/30/2010 12:03:49
  3 ORIOLES  Down (not IPLed)
  4 ----- Available
  5 ----- Available
  6 ----- Available
  7 ----- Available
  8 ----- Available
```

- Attempting to add a member to a slot while in REPAIR mode results in the following response:

```
set ssi slot 3 orioles
HCP1658E SET SSI command is not allowed because this system was
IPLed with the REPAIR IPL parameter.
```

- Querying the current SSI cluster displays the following information:

```
query ssi
SSI Name:  ALEAST
SSI Mode:  Stable
SSI Persistent Data Record (PDR) device: SSIVOL on 3333
SLOT SYSTEMID STATE      PDR HEARTBEAT      RECEIVED HEARTBEAT
  1 YANKEES  Joined    01/30/2015 12:03:33 01/30/2015 12:03:35
```

```

2 REDSOX   Joined    01/30/2015 12:03:47 01/30/2015 12:03:49
3 ----- Available
4 ----- Available
5 ----- Available
6 ----- Available
7 ----- Available
8 ----- Available

```

To replace the current PDR volume SSIVOL with new PDR volume NEWVOL, enter:

```
set ssi pdrvvolume NEWVOL
```

Subsequently, querying the updated SSI cluster displays the following information:

```

query ssi
SSI Name:  ALEAST
SSI Mode:  Stable
SSI Persistent Data Record (PDR) device: NEWVOL on 4444
SLOT SYSTEMID STATE      PDR HEARTBEAT      RECEIVED HEARTBEAT
  1 YANKEES   Joined    01/30/2015 12:03:33 01/30/2015 12:03:35
  2 REDSOX   Joined    01/30/2015 12:03:47 01/30/2015 12:03:49
  3 ----- Available
  4 ----- Available
  5 ----- Available
  6 ----- Available
  7 ----- Available
  8 ----- Available

```

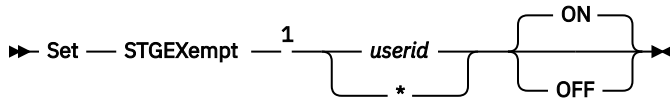
## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP1640E Volume *valid* is already the PDR volume
- HCP1640E Replacement PDR volume *valid* is not properly formatted
- HCP1640E Replacement PDR volume *valid* is not online
- HCP1640E Replacement PDR volume *valid* is not attached to the system
- HCP1640E SET SSI PDRVOLUME feature is not available on all member systems
- HCP1640E SET SSI PDRVOLUME was rejected on system *sysname*
- HCP1643E System *sysname* was specified more than once.
- HCP1645E SET SSI command did not complete because an error occurred reading the PDR
- HCP1648E SET SSI command did not complete because a higher priority SSI action is in progress.
- HCP1649E SET SSI command was rejected by another member of the SSI cluster.
- HCP1650E SET SSI command is not allowed because SSI mode is *mode*.
- HCP1651E SET SSI command is not allowed because this system's state is Isolated.
- HCP1652E SSI system slot *slot\_number* is already Available.
- HCP1653E System slot *slot\_number* cannot be marked Available because its current state is *state*.
- HCP1654E SSI system slot *slot\_number* already in use.
- HCP1655E System identifier already being used in slot *slot\_number*.
- HCP1656E System *sysname* cannot be marked Down because its current state is *state*.
- HCP1657E System *sysname* cannot be marked Down because it is currently active.
- HCP1658E SET SSI command is not allowed because this system was IPLed with the REPAIR IPL parameter
- HCP1659E SET SSI command is not allowed because the user directory is not compatible with SSI.
- HCP1664E SET SSI command is not allowed because { the specified system | system *sysname* } is not a member of the SSI cluster.
- HCP1666E This system is not a member of an SSI cluster.

## SET SSI

- HCP1668E System *sysname* cannot be forced Down because it is the local system.
- HCP1816E A relocation domain with the name *domain\_name* has already been defined
- HCP6650E Slots above *slot* cannot be defined because the cluster SPOOL\_MEMBERS setting on the SSI\_CONTROLS statement is set to *slot*.
- HCP6704E Missing token at end of line
- HCP6706E Invalid number - *number*
- HCP6706E Invalid volume label - *valid*
- HCP6706E Invalid system name - *sysname*

## SET STGEXEMPT



Notes:

<sup>1</sup> If SET STGEXEMPT is executed with the AT command, the *userid* operand is required.

### Authorization

Privilege Class: A, B, C

### Purpose

Use SET STGEXEMPT to control the state of free storage limit detection for virtual machines.

### Operands

#### *userid*

specifies the user ID for which free storage limit detection is to be disabled or enabled.

\*

specifies that free storage limit detection is to be disabled or enabled for the user issuing the command.

#### ON

specifies that the user ID is exempt from free storage limit detection.

#### OFF

specifies that the user ID is subject to free storage limit detection.

### Usage Notes

1. The SET STGEXEMPT command sets the state of free storage limit detection for the user ID. If you exempt a user ID from free storage limit detection, usage of CP storage in support of the user ID is unbounded. If you subject the user ID to free storage limit detection, CP will detect if too much CP storage is consumed in support of the user ID and if so, the user ID will be stopped or possibly forced off the system.

### Responses

Response 1:

```

Userid USERID is
{exempt from} free storage limit detection
{subject to }
  
```

is the response if the specified user ID was not previously {exempt from|subject to} free storage limit detection. There may be an additional response saying that free storage limit detection is off for the system.

Response 2:

```

Userid USERID is already
  
```

## SET STGEXEMPT

```
{exempt from} free storage limit detection  
{subject to }
```

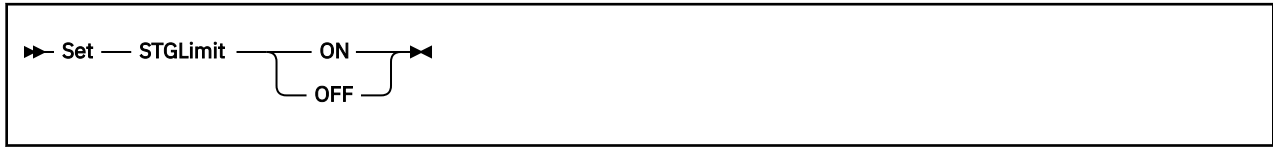
is the response if the specified user ID was previously {exempt from|subject to} free storage limit detection. There is an additional response given if free storage limit detection is off for the system.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on



## SET STGLIMIT



### Authorization

Privilege Class: A, B, C

### Purpose

Use SET STGLIMIT to control the state of free storage limit detection for the system.

### Operands

#### ON

specifies that free storage limit detection is to be turned on for the system. CP will detect if too much CP free storage is consumed in support of userids and if so, those userids consuming excessive free storage will be stopped or possibly forced off the system.

#### OFF

specifies that free storage limit detection is to be turned off for the system. Usage of CP free storage in support of userids is unbounded.

### Responses

Response 1:

```
Free storage limit detection is now {ON }
                                   {OFF }
```

is the response if free storage limit detection was not previously {ON|OFF}.

Response 2:

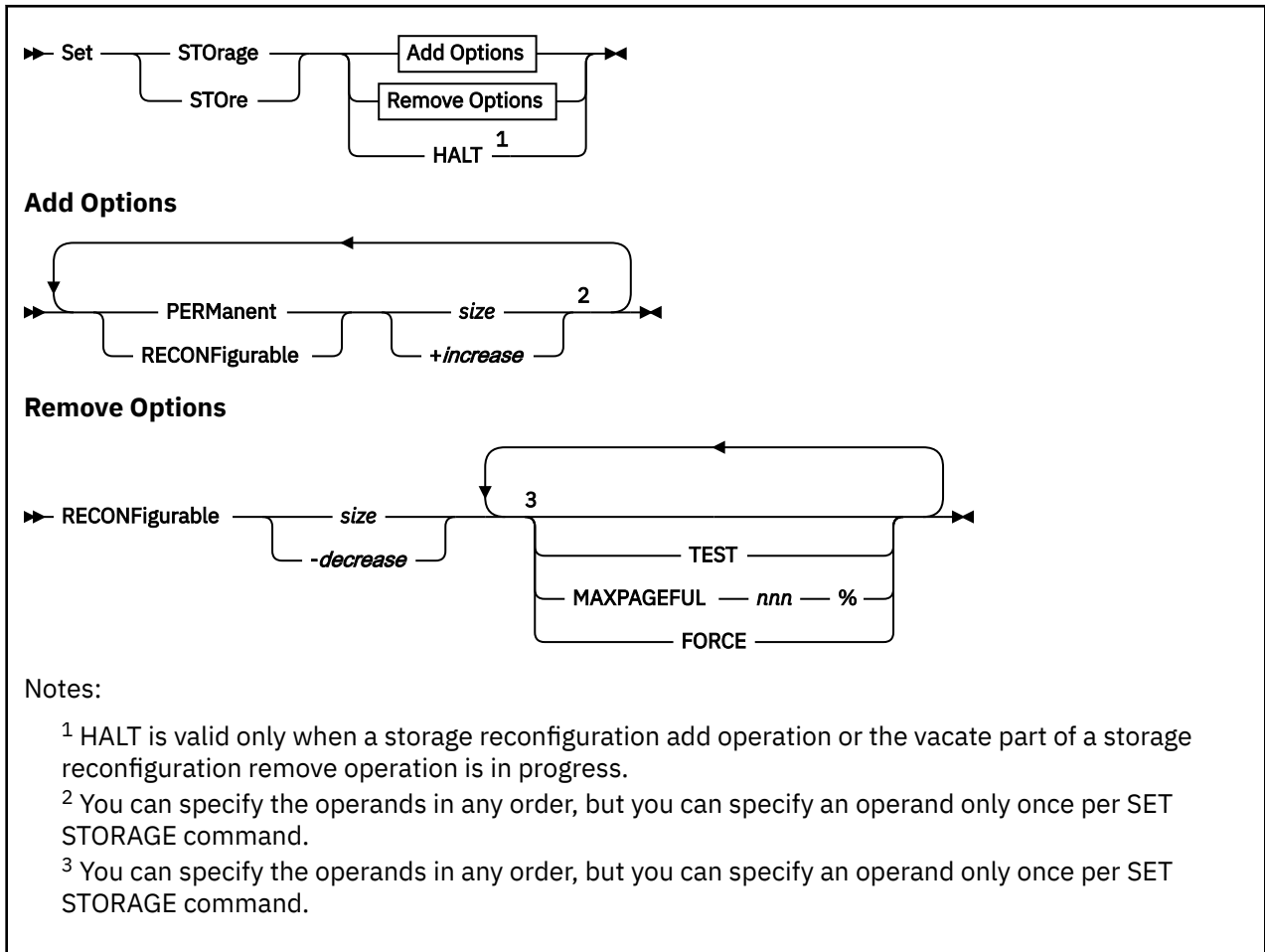
```
Free storage limit detection is already {ON }
                                       {OFF }
```

is the response if free storage limit detection was previously {ON|OFF}.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP0026E Operand missing or invalid

## SET STORAGE



### Authorization

Privilege Class: A

### Purpose

Use SET STORAGE or SET STORE to dynamically increase or decrease the amount of real storage used by z/VM. No system reIPL is required. To be removable, storage must be defined as reconfigurable when it is added to the z/VM configuration. Any increase or decrease in the size of the configuration must be done in multiples of the increment size, as identified in the response to the QUERY STORAGE command.

### Operands

#### PERManent

identifies real storage that cannot be removed from the z/VM configuration.

#### RECONFigurable

identifies real storage that can be removed from the z/VM configuration.

#### size

is the new size of online real storage of the type specified: permanent or reconfigurable. The amount of reconfigurable storage cannot exceed 50% of all online storage. The *size* value has the format *nu*, where *n* is a 1- to 7-digit decimal number that is greater than 0 and *u* is the 1-character storage unit suffix (see [Table 38 on page 1723](#)). The value you specify must be a multiple of the increment size. A

value of 0, without a storage size suffix, is allowed when the memory type is RECONFIGURABLE and you want to remove all reconfigurable storage.

If your system is using SET RESERVED commands, it is recommended you explicitly set the system maximum (via the STORAGE RESERVED SYSMAX statement in the SYSTEM CONFIG file or by using the SET RESERVED SYSMAX command) to a value that is less than the size of permanent storage. This is particularly important when you are planning to reclaim all memory from reconfigurable storage, to make sure the total reserved storage on the system does not exceed the size of permanent storage.

To increase the amount of storage that is in use by z/VM, specify a new permanent or reconfigurable storage size that is greater than the current amount of storage for that type. The PERMANENT and RECONFIGURABLE operands can be specified on the same SET STORAGE command only when you are increasing the amount of each type of storage.

To decrease the amount of storage that is in use by z/VM, specify a new reconfigurable storage size that is less than the current amount of reconfigurable storage.

Table 38. Maximum Input Values for Storage Units

Storage unit (powers of 2)	Suffix ( <i>u</i> )	Maximum input value ( <i>n</i> )
Megabytes	M	9999999
Gigabytes	G	9999999
Terabytes	T	9999999
Petabytes	P	16384
Exabytes	E	16

**Note:** The maximum input value of 9999999 for the M, G, or T suffix is not a size limit; it is the physical limit of the operand (7 digits plus suffix). If the maximum input value for one of these suffixes does not allow you to define the amount of storage you want, you need to use a larger storage unit.

#### **+increase**

is the amount of permanent or reconfigurable real storage to add to the current z/VM configuration. The value by which to increase storage has the format *nu*, where *n* is a 1- to 7-digit decimal number greater than 0 and *u* is the 1-character storage unit suffix (see [Table 38 on page 1723](#)).

The value you specify must be a multiple of the increment size.

#### **-decrease**

is the amount or reconfigurable real storage to remove from the current z/VM configuration. The value by which to decrease storage has the format *nu*, where *n* is a 1- to 7-digit decimal number greater than 0 and *u* is the 1-character storage unit suffix (see [Table 38 on page 1723](#)).

The value you specify must be a multiple of the increment size.

#### **TEST**

determines the viability of a storage removal by ensuring that, if the specified amount of reconfigurable storage was removed, there are adequate amounts of real storage frames and paging slots to support the current workload. No actual removal of reconfigurable storage occurs.

#### **MAXPAGEFULL *nnn*%**

specifies the upper bound on the number of paging slots in use as a percentage of the total number of slots on all paging volumes. The *nnn* value is a decimal integer in the range of 0 to 100. This value is used in the storage reclamation viability checks performed during execution of this SET STORAGE command to make sure that the removal of storage does not cause the current workload to exceed the available space in the storage hierarchy (real storage plus paging space). The viability check is performed when the TEST operand is specified, and during a storage reclamation process (at the start and periodically throughout the reclamation).

When a MAXPAGEFULL value is not specified, the default value used is the warning percentage specified on the PAGING system configuration statement or on a previous SET PAGING command.

## SET STORAGE

Specification of the MAXPAGEFULL parameter on a SET STORAGE command does not alter the warning value established by the PAGING system configuration statement or a previous SET PAGING command.

### FORCE

indicates that no viability check should be performed prior to, or during, a storage reclamation. Use this operand only when you have determined that the storage reclamation will not cause the current workload to exceed the storage resources in your configuration. See Usage Note “10” on page 1726.

### HALT

initiates termination of a storage reconfiguration in progress. The reconfiguration operation is terminated asynchronously from the command invocation. Any portion of the reconfiguration already completed is not undone. A HALT issued during a storage reclamation only affects the vacate portion of the operation. If the storage being reclaimed is vacated and all of the removed storage has not yet been transitioned to standby, the halt will fail.

## Usage Notes

1. When you define a logical partition, two storage allocations, `Initial` and `Reserved`, are specified in the image profile for the partition. The maximum amount of storage that can be used by an operating system running in the partition is the sum of these two storage allocations. The machine uses the `Initial` and `Reserved` amounts in determining the storage increment size. The storage increment size is the unit in which storage can be added to or removed from the configuration.

For a fresh IPL (one that is not an automatic restart), the following considerations apply:

- CP uses the `PERMANENT` and `RECONFIGURABLE` specifications on the `STORAGE` statements in the system configuration file to determine how to initialize storage. There must be at least 4 GB of permanent storage online before any reconfigurable storage can be configured.
- Alternatively, you can use the IPL parameter `STORE=n` to override the storage configuration information in the system configuration file. In this case, all of the storage specified via `STORE=n` is initialized as permanent.
- If neither the `STORAGE` statement nor the IPL parameter `STORE=` was specified, all of the storage z/VM finds configured in the partition is initialized as permanent storage. If this is the first IPL after the logical partition was activated, the amount of configured storage equals the `Initial` storage allocation. If this is a subsequent IPL, the amount of storage configured at IPL depends on the configuration left by the operating system previously running in the partition.

The storage values on the `STORAGE` statement and the IPL parameter `STORE=n` are rounded up to a multiple of the storage-increment size, if necessary.

For an IPL that is the result of an automatic restart, the `AFTER_RESTART` and `AFTER_SHUTDOWN_REIPL` operands on the `STORAGE` statement in the system configuration file determine z/VM's behavior in establishing the storage configuration. The `AFTER_RESTART` operand applies to an automatic reIPL resulting from a CP abend or a PSW restart. The `AFTER_SHUTDOWN_REIPL` operand applies to an automatic reIPL resulting from the `SHUTDOWN` command with the `REIPL` operand. Parameters on these operands provide the ability for you to initialize the storage configuration as though it were a fresh IPL, or to keep the storage configuration as it was at system termination prior to the reIPL. See the [STORAGE Statement](#) in *z/VM: CP Planning and Administration* for more information.

2. After IPL, you can perform a storage reconfiguration using the SET STORAGE command to bring more permanent or reconfigurable storage online to z/VM, or to take some or all of the reconfigurable storage offline. The following fields displayed by the QUERY STORAGE command with the `RECONFIGURABLE` operand can guide you in determining the extent to which storage is reconfigurable.
  - `STORAGE=` is the amount of real storage currently online to z/VM and is the sum of the `Permanent=` and `Reconfigurable=` values. Additional permanent or reconfigurable storage can be brought online, up to the `Maximum STORAGE=` value displayed. This maximum is the sum of the `Initial`

- and Reserved settings for the logical partition in the image profile, or the maximum amount of real storage this z/VM system supports, whichever is less.
- The sum of CONFIGURED=, STANDBY=, and RESERVED equals the sum of the Initial and Reserved storage allocations in the image profile for the logical partition, or the z/VM-supported maximum, whichever is less.
  - Any storage you add or remove must be specified in multiples of the storage increment size as indicated by the INC= value.
  - Additional storage brought online to z/VM is obtained from standby storage (STANDBY=). The amount of standby storage at any time is determined by the machine, based on the amount of installed storage that is not currently claimed by other active logical partitions in the CPC, along with the Initial and Reserved storage allocations in the image profile for the partition. Although the sum of Initial and Reserved storage allocations in the image profile can exceed the z/VM-supported maximum, the STANDBY= value never allows the system to grow beyond the z/VM-supported maximum. To move storage from the RESERVED= state to the STANDBY= state, manual actions can be taken, such as deactivating another logical partition or installing more real storage.
  - Any amount of Reconfigurable= storage can be removed from the z/VM configuration. The removed storage will increase the STANDBY= value, the RESERVED= value, or both.
3. After a storage reconfiguration has begun, the QUERY STORAGE command with the RECONFIGURABLE operand can be used to display the status of the reconfiguration, including the amount of storage remaining to be added or removed, and the elapsed time of the reconfiguration.
  4. To increase the likelihood of a successful storage add operation, the following steps are recommended:
    - a. Make sure all other storage reconfigurations have completed. This includes logical partition activation or deactivation, as well as storage configuration actions by other operating systems in the central processor complex (CPC).
    - b. Issue the QUERY STORAGE command as necessary to make sure the reported STANDBY= and RESERVED= values are stable.
    - c. Make sure that available standby storage is equal to or greater than the amount of storage you want to add to the z/VM configuration.
    - d. After completing the previous steps, issue the SET STORAGE command to increase the size of the z/VM configuration.
  5. When adding storage, the final step in the storage initialization process can be performed only during nominal system activity. Therefore, all of the storage added might not appear to be available immediately. The NotInitialized= value displayed in the QUERY FRAMES command response is the count of uninitialized frames. The command issuer receives an asynchronous message when all storage is brought online and initialized. No other SET STORAGE commands are allowed until storage initialization is complete. The NotInitialized= value displayed in the QUERY FRAMES command response is the count of uninitialized frames.
  6. After increasing real storage you might want to evaluate and increase the storage settings that you specified for SET RESERVED, SET MDCACHE STORAGE, and the SET VDISK system limit.
  7. Because the SET STORAGE command can be used to remove reconfigurable storage at any time, z/VM limits allocations within reconfigurable storage to data that can be easily moved. To provide the most flexibility to z/VM in allocating memory, you should limit the amount of reconfigurable storage you define to be the amount you expect to eventually remove. When there is an insufficient amount of permanent storage available for normal system operation, z/VM converts a reconfigurable storage increment to permanent storage. When this happens, message HCP2594I is issued to the system operator.
  8. The amount of time needed to complete a reconfiguration to remove storage depends on a number of factors, including: the amount of storage to remove, the state of the guest pages within the storage, the availability of frames in which to move page content, and the paging bandwidth. You can issue the QUERY STORAGE command with the RECONFIGURATION operand to display the status of a storage

## SET STORAGE

removal in progress. The command issuer receives an asynchronous message when all storage is vacated and taken offline (unassigned).

9. Stabilizing the memory usage of your workload before removing reconfigurable storage enables z/VM to more accurately assess the storage demands on the system prior to any storage removal, and throughout the removal operation.

The reclamation viability check, as described under the MAXPAGEFULL operand, is performed prior to the removal of any reconfigurable storage and also intermittently throughout the storage removal process, unless the FORCE operand is specified. If at any time the viability check fails, the system does not remove any additional storage and message HCP2650E is issued to the user who initiated the storage removal and (if a different person) the system operator.

10. If you specify or default to AGELIST KEEPSLOT YES in the system configuration file, pageable pages can have copies in real storage and also on paging volumes. In this case, the reclamation viability check might be overly conservative, because it does not consider the number of guest pages having two copies in the storage hierarchy. If the amount of unused real storage is less than the amount of storage being removed, a storage reclamation will push pages out to DASD. The viability estimation assumes every page written to DASD is not already associated with a slot, and therefore will increase the amount of paging space in use. Depending on the amount of paging space currently in use in your configuration, this could result in a failed viability check and an unsuccessful storage reclamation.

If you have received a failed viability check and are considering using the FORCE option, you must first consider the full storage situation on your system, including real storage, paging capacity and usage, and virtual storage (for example: instantiated, resident, and DASD page counts for users, as well as writeable NSS/DCSS pages, CP pageable pages, and CP control structures). The VIR2REAL EXEC and the CHKRECLM EXEC can be helpful in assessing your storage usage and determining if the FORCE option is appropriate. The VIR2REAL EXEC gathers information about the real and virtual storage on your system as it currently appears (that is, based on the guests that are logged on and the current paging usage). The CHKRECLM EXEC accepts a storage amount as input, and uses information about your current storage configuration and usage to model how your system might look after a reclamation. It shows how the reclamation could change the virtual to real ratios, real storage, and paging. These EXECs are available here:

[VM Download Packages \(https://www.vm.ibm.com/download/packages/\)](https://www.vm.ibm.com/download/packages/)

## Responses

Response 1:

When a SET STORAGE command is issued to add storage, the command issuer and the system operator receive the following messages when the operation begins and ends:

```
HCP2580I Storage reconfiguration to add permsize of permanent and reconfsize
of reconfigurable initiated by userid.
.
HCP2582I Storage reconfiguration by userid is complete.
Permanent = permsize Reconfigurable = reconfsize
```

where:

### ***permsize***

is a storage amount represented by an integer followed by a storage unit suffix (see [Table 38 on page 1723](#)).

### ***reconfsize***

is a storage amount represented by an integer followed by a storage unit suffix (see [Table 38 on page 1723](#)).

### ***userid***

is the user ID that issued the command.

Response 2:

When a SET STORAGE command is issued to remove storage, and CP determines the reclamation will not fill paging volumes beyond the MAXPAGEFULL percentage, the command issuer receives the following message:

```
HCP2593I Storage reclamation viability test passed with MAXPAGEFULL value of
xxx%. yyy% of paging space could be required for the current workload.
```

where:

**xxx%**

is the maximum percent of paging slots in use that you will tolerate for this reclamation. This is the value of the MAXPAGEFULL operand specified on the SET STORAGE command or, if MAXPAGEFULL was not specified on the command, it is the PAGING WARNING value specified via the PAGING system configuration statement or a previous SET PAGING command.

**yyy%**

is CP's estimate for the maximum amount of paging space in use after the storage removal completes.

If the TEST operand is not specified, the storage removal process is started. The command issuer and the system operator (if a different person) receive the following messages when the storage reconfiguration remove operation begins and ends:

```
HCP2581I Storage reconfiguration to remove reconfsize of reconfigurable
initiated by userid.
:
HCP2582I Storage reconfiguration by userid is complete.
Permanent = permsize Reconfigurable = reconfsize
```

where *permsize*, *reconfsize*, and *userid* are as described in Response 1.

Response 3:

When a SET STORAGE command is issued to remove storage and CP determines that the reclamation will fill paging volumes beyond the MAXPAGEFULL percentage, no storage removal operation is initiated and the command issuer receives the following message.

```
HCP2650E Storage reclamation viability test failed with MAXPAGEFULL value of
xxx%. yyy% of paging space could be required for the current workload.
```

where *xxx%* and *yyy%* are as described in Response 2.

Response 4:

When a SET STORAGE HALT command is issued, the command issuer, the system operator, and the user who initiated the storage reconfiguration (unless they are all the same person) receive the following messages when the storage reconfiguration has been signaled to stop processing and when the storage reconfiguration ends.

```
HCP2583I Storage reconfiguration has been signaled to halt processing by
userid.
:
HCP2584I Storage reconfiguration has been halted by userid.
```

where *userid* is as described in Response 1.

## Messages

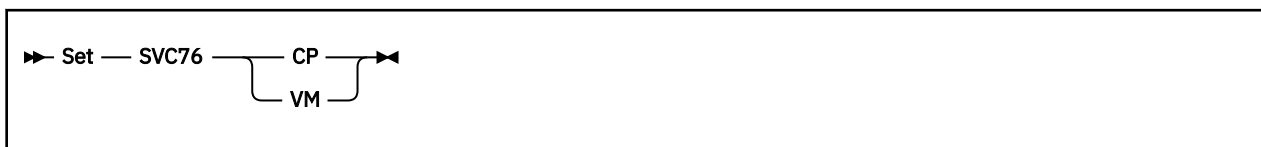
- HCP003E Invalid option - *operand*
- HCP013E Conflicting option - *operand*
- HCP025E Storage missing or invalid
- HCP026E Operand missing or invalid

## SET STORAGE

- HCP097E STANDBY storage of *stbysize* is insufficient to satisfy request
- HCP100E Storage reconfiguration is active. You cannot change storage at this time.
- HCP1133I Insufficient STANDBY storage to complete storage {initialization|reconfiguration} as requested.
- HCP2576E The command processing cannot complete and a soft abend has been taken because an unrecoverable system error has occurred.
- HCP2579E SET STORAGE command unavailable [until system reIPL]
- HCP2580I Storage reconfiguration to add *permsize* of permanent and *reconfsize* of reconfigurable initiated by *userid*.
- HCP2581I Storage reconfiguration to remove *reconfsize* of reconfigurable initiated by *userid*.
- HCP2582I Storage reconfiguration by *userid* is complete. Permanent = *permsize* Reconfigurable = *reconfsize*
- HCP2583I Storage reconfiguration has been signaled to halt processing by *userid*.
- HCP2583I Storage reconfiguration halt was already initiated by {*userid*|SYSTEM}.
- HCP2584I Storage reconfiguration has been halted by {*userid*|SYSTEM}.
- HCP2585E Storage request exceeds Maximum STORAGE of *maxsize*.
- HCP2585E Reconfigurable storage size of *stgsize* exceeds the maximum allowed reconfigurable storage size of *maxsize*. Reconfigurable storage is restricted to a maximum of *xx%* of all online storage.
- HCP2586E RECONFIGURABLE storage of *reconfsize* is insufficient to satisfy request.
- HCP2587E PERMANENT storage of *permsize* must be online before RECONFIGURABLE storage can be added to the configuration.
- HCP2588E RECONFIGURABLE storage is not supported on this system.
- HCP2589E Storage size *storsize* is not a multiple of the storage increment size *incrsz*. The nearest increment multiples are *mult1* and *mult2*.
- HCP2590E SET STORAGE HALT cannot be processed because there is no user-initiated storage reconfiguration in progress.
- HCP2591E Storage reconfiguration failed - Processor controller response code *xxx*
- HCP2593I Storage reclamation viability test passed with MAXPAGEFULL value of *xxx%*. *yyy%* of paging space could be required for the current workload.
- HCP2598E PERMANENT storage cannot be added and RECONFIGURABLE storage removed in the same SET STORAGE command.
- HCP2599E SET STORAGE HALT cannot be processed because the storage has been vacated.
- HCP2650E Storage reclamation viability test failed with MAXPAGEFULL value of *xxx%*. *yyy%* of paging space could be required for the current workload.



## SET SVC76



### Authorization

Privilege Class: G

### Purpose

Use SET SVC76 to specify whether the control program (CP) or your virtual machine will handle the SVC76 instruction.

### Operands

#### SVC76 CP

#### SVC76 VM

controls whether SVC76 instructions issued by your virtual machine are processed by CP for recording of virtual device errors. Specify CP if guest error records are to be recorded by CP when your virtual machine is running in supervisor state. Specify VM if you do not want errors recorded and if you want all guest SVC76 operations to be processed by your virtual machine. At logon, SVC76 is set to VM if the SVC76VM parameter is specified in the OPTION control statement, or to CP if the SVC76VM parameter is not specified in the OPTION control statement.

### Usage Notes

1. If your virtual machine has SET SVC76 set to CP, then SVC76 is reflected to the user, and does one of the following:
  - Issues an SVC76 from problem state
  - Uses a not valid SVC76 interface
  - Presents an unrecognizable EREP record
  - Presents an EREP record that cannot be translated.
2. Unless you are intentionally simulating hardware errors in your virtual machine (for example, in order to test error recovery procedures), you should let CP record all errors. When CP maintains the hardware error log, your installation has the most accurate information about equipment errors.
3. Linux does not use SVC76 to record hardware errors. Therefore, if the guest has explicitly identified itself to z/VM as running Linux, then the setting of this command is ignored and it is treated as if the setting is always VM.

### Messages

- HCP003E Invalid option - {*option*|command contains extra options starting with *option*}
- HCP026E Operand missing or invalid

## SET SYSCONTROL

---

►► Set — SYSCONTROL — options ◄◄

### Authorization

Privilege Class: A

### Purpose

Use SET SYSCONTROL command to change certain system behaviors only when advised to do so by IBM support personnel. These behaviors are not related to system integrity or security policy, are subject to change without notice, and are not further documented.

## SET SYSOPER



### Authorization

Privilege Class: A

### Purpose

Use SET SYSOPER to change the user ID of the primary system operator on your system.

### Operands

#### *userid*

is the user ID that is to become the active operator. The specified user must:

- Be logged on or disconnected
- Possess at least one of the privilege classes required for system operators, which are defined on the PRIV\_CLASSES statement in the system configuration file, or must be the special operator user ID defined on the SYSTEM\_USERIDS statement.

\*

represents the user ID of the virtual machine issuing the command. The virtual machine must possess at least one of the privilege classes required for system operators, which are defined on the PRIV\_CLASSES statement in the system configuration file.

### Usage Notes

1. Use the QUERY SYSOPER command to display the user ID of the current primary system operator.  
For more information, see [“QUERY SYSOPER” on page 1202](#).
2. If the target virtual machine's user ID is specified as the system operator ID on the SYSTEM\_USERIDS statement in the system configuration file, it need not have any special privilege classes.
3. You can also change the primary system operator by logging off the current primary system operator and logging on the desired operator ID. As with the SET SYSOPER command, the new primary operator must either possess the required privilege classes or must be the special operator user ID specified on the SYSTEM\_USERIDS statement in the system configuration file.
4. An alternative to the above is to use the ALTERNATE\_OPERATORS statement in the system configuration file. The ALTERNATE\_OPERATORS statement lets you specify a list of user IDs that can become the system operator automatically when the primary system operator logs off. For more information on using the [ALTERNATE\\_OPERATORS Statement](#), see *z/VM: CP Planning and Administration*.
5. The primary system operator receives all messages sent to the OPERATOR user ID.

### Responses

Response 1:

The command issuer receives no response. CP sends the following message to the new operator user ID:

```
HCPCF06776I userid selected your userid
as the system operator ID.
```

## SET SYSOPER

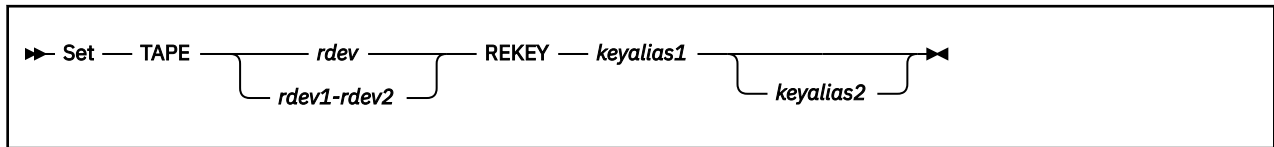
CP sends the following message to the old operator user ID:

```
HCPCF06776I userid removed your userid  
as the system operator ID.
```

### Messages

- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP007E Invalid *userid* - *userid*
- HCP020E *Userid* missing or invalid
- HCP045E *userid* not logged on
- HCP6777E *userid* in IPL, LOGOFF, or FORCE process
- HCP6778E User *userid* does not have required privilege class.

## SET TAPE



### Authorization

Privilege Class: B

### Purpose

Use SET TAPE to reencrypt (rekey) an encrypted tape with different key information without rewriting the tape.

### Operands

*rdev*

*rdev1-rdev2*

specifies a real device number, a list of real device numbers, or a range of real device numbers.

**REKEY** *keyalias1*

**REKEY** *keyalias2*

reencrypts a tape has was already encrypted using different key information so that you do not need to rewrite the tape. At least one *keyalias* must be specified.

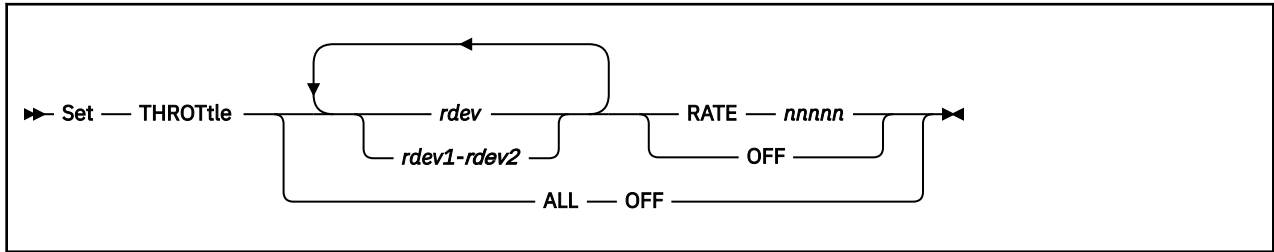
### Usage Notes

1. The range of devices that are specified on a single CP SET TAPE command cannot span more than 256 real device numbers.
2. The REKEY function is only valid when issued against 3592 tape drives that support hardware encryption and rekeying, and when the mounted tape cartridge is encrypted and positioned at the beginning of the tape (BOT). This will take a few moments to complete as it performs the necessary key exchanges between the control unit and the key manager.

### Messages

- HCP006E Invalid device type - *rdev*
- HCP040E Device *rdev* does not exist
- HCP1130E Device *rdev* does not support a rekey operation
- HCP1131I Rekey operation succeeded for device *rdev*
- HCP1132E Rekey operation failed for device *rdev*.
- HCP1132E Rekey operation failed for device *rdev*. Tape cartridge is invalid.
- HCP6000E The range of device numbers cannot exceed 256.
- HCP9968E Key alias not found: *keyalias*.

## SET THROTTLE



### Authorization

Privilege Class: B

### Purpose

Use SET THROTTLE to control the number of I/O operations that a guest operating system can initiate to a specific real device. This prevents a guest from interfering with or dominating I/O resources.

Using the SET THROTTLE command, you can either:

- Limit the guest's I/O rate to a specific I/O device by adding that device to the I/O throttling list, or
- Give the guest and your first-level system equal control over the specified I/O device by removing that device from the I/O throttling list.

### Operands

***rdev* RATE *nnnnn***

***rdev1-rdev2* RATE *nnnnn***

adds one or more real devices to the I/O throttling list at the specified I/O rate or changes the I/O rate of one or more devices currently in the I/O throttling list. After adding devices to or changing I/O rates in the list, CP immediately begins limiting (throttling) the guest's I/O rate to the specified device or devices.

The variable *rdev* is the real device number of the device that you want to add to or change in the I/O throttling list. Each *rdev* must be a 1- to 4- digit hexadecimal number between X'0000' and X'FFFF'. You can specify a single device address, a range of device addresses, or any combination thereof.

The variable *nnnnn* is I/O throttling rate (number of I/O operations per second that this device can process from a guest operating system). It must be a decimal number between 1 and 10,000.

**ALL OFF**

tells CP to remove all the devices from the I/O throttling list.

***rdev* OFF**

***rdev1-rdev2* OFF**

tells CP to remove the specified device or devices from the I/O throttling list. The variable *rdev* is the real device number of the device that you want to stop throttling. It must be a hexadecimal number between X'0000' and X'FFFF'. You can specify a single device address, a range of device addresses, or any combination thereof.

### Usage Notes

1. CP can throttle devices that are eligible for I/O assist, but not while the device is in I/O assist. When a device is in I/O assist, the guest is controlling the device directly, not CP. CP can only throttle a device when it controls the I/O to and from that device. Thus, if you throttle a device that is currently in I/O assist, CP accepts your command, but does not actually begin throttling the device until the device is no longer in I/O assist.

2. To display which devices are on the I/O throttling list and what their rates are, use the QUERY THROTTLE command (see “[QUERY THROTTLE](#)” on page 1217).
3. You can also add devices to the I/O throttling list using the THROTTLE statement in the system configuration file. For more information about the [THROTTLE Statement](#), see *z/VM: CP Planning and Administration*.

## Responses

Response 1:

To begin throttling a range of devices, enter the following:

```
set throttle 800-804 rate 10000
23:08:36 HCPTHR040E Device 0801 does not exist
23:08:36 HCPTHR040E Device 0804 does not exist
23:08:36 3 device(s) added to I/O throttling list starting with device 0800
23:08:36 2 device(s) skipped. Rate is 10000 I/Os per second
```

Response 2:

To stop throttling all devices, enter the following:

```
set throttle all off
23:09:32 The I/O throttling list has been cleared
```

Response 3:

To stop throttling a range of devices, enter the following:

```
set throttle 800-804 off
23:10:36 HCPTHR040E Device 0801 does not exist
23:10:36 HCPTHR040E Device 0804 does not exist
23:10:36 3 device(s) deleted from the I/O throttling list starting
with device 0800
23:10:36 2 device(s) skipped
```

Response 4:

To begin throttling a device, enter the following:

```
set throttle 800 rate 10000
23:25:24 Device 0800 added to I/O throttling list. Rate 10000 I/Os
per second
```

Response 5:

To stop throttling a device, enter the following:

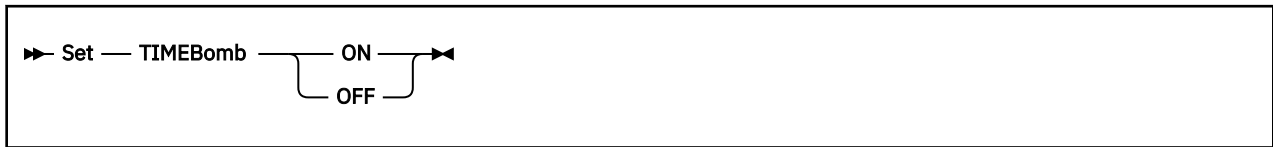
```
set throttle 800 off
23:25:36 Device 0800 deleted from the I/O throttling list
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP040E Device *rdev* does not exist
- HCP1150E DASD *rdev* is not a valid base exposure.
- HCP6810E Device *rdev* is in use by the system and cannot be throttled
- HCP6810E Device *rdev* is not allowed to be throttled
- HCP6810E Device *rdev* is not in the I/O throttling list
- HCP6810E Device *rdev* is offline and cannot be throttled
- HCP6810E Device *rdev* is reserved and cannot be throttled

## SET TIMEBOMB

---



### Authorization

Privilege Class: G

### Purpose

Use SET TIMEBOMB to turn on and off the dynamic virtual machine time bomb. The time bomb is a mechanism for allowing a virtual machine to cause an action to be performed if it becomes unresponsive.

### Operands

**ON**

**OFF**

ON specifies that the virtual machine time bomb facility is to operate normally. OFF specifies that the virtual machine time bomb facility is to operate normally except that the time bomb is disabled.

### Usage Notes

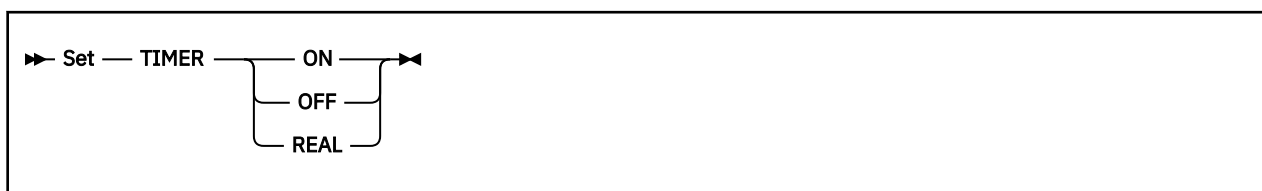
1. With the time bomb disabled, detection of an unresponsive condition does not cause the associated action to be performed. In all other respects, the time bomb facility appears to the virtual machine to be operating normally.
2. It may be convenient to disable the time bomb when debugging a virtual machine application, where delays caused by the debugging functions may exceed the time bomb timeout interval.
3. An application establishes a time bomb using DIAGNOSE Code X'288' - Control Virtual Machine Time Bomb, see z/VM: CP Programming Services for more information.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid



## SET TIMER



### Authorization

Privilege Class: G

### Purpose

Use SET TIMER to control the interval timer for your virtual machine. SET TIMER is valid only if SET 370ACCOM is ON.

### Operands

#### ON

specifies that the virtual timer is updated only when the virtual processor is running. The initial setting when you log on is ON.

#### OFF

specifies that the timer is not updated.

#### REAL

specifies that the timer is updated during the virtual processor run time and also during virtual wait time.

### Usage Notes

1. If the 370 Accommodation facility is active the interval timer is present for XA, ESA, and XC virtual machines. Use the CP SET 370ACCOM ON command to activate the 370 Accommodation facility.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid

## SET TIMEZONE

```
►► Set — TIMEZONE — zoneid ◄◄
```

### Authorization

Privilege Class: A

### Purpose

Use SET TIMEZONE to change the system's active time zone ID and time zone offset.

**Note:** We strongly recommend you enter this command only when changing to standard or daylight saving time.

### Operands

#### *zoneid*

*zoneid* is the time zone ID that you want to make active, such as EST. To get a list of the defined time zones, enter a QUERY TIMEZONES command.

### Usage Notes

1. Some applications may be adversely affected by changes in time zones. IBM recommends that you stop these applications before changing the time zone and start them again after the time zone has been changed.
2. Even if the new time zone ID is the same as the current time zone ID, a privileged user may use the DEFINE TIMEZONE command to change the offset associated with the time zone ID. To put the new offset into effect, you must enter the SET TIMEZONE command. CP will issue all messages and generate all external interrupts associated with a normal time zone change. For more information on defining a new time zone or changing the existing time zone definition, see the DEFINE TIMEZONE command.
3. You can tell CP whether you want your virtual machines to receive a special external interrupt when a time zone change occurs. See the description of [DIAGNOSE Code X'274' - Set Timezone Interrupt Flag](#) in *z/VM: CP Programming Services* for details.
4. CMS levels prior to CMS level 11 do not request time zone notification using DIAGNOSE code X'274', therefore, file system time stamps do not reflect the change in time zone unless you re-IPL CMS.
5. You can check the current time zone by issuing the QUERY TIMEZONES command.
6. You can find the difference between your system's current time zone and the Coordinated Universal Time (UTC) by issuing the QUERY TIME command with the OFFSET operand.
7. Accounting records continue to contain local date and time values.
8. Queued timer interrupts are based on the Time of Day Clock, which is not changed by a SET TIMEZONE command.
9. This command is rejected when CP is deriving its time zone information from an external source (for example, when the STP\_TZ feature is enabled from the configuration file).

### Responses

Response 1:

When you enter the SET TIMEZONE command, CP issues the following response:

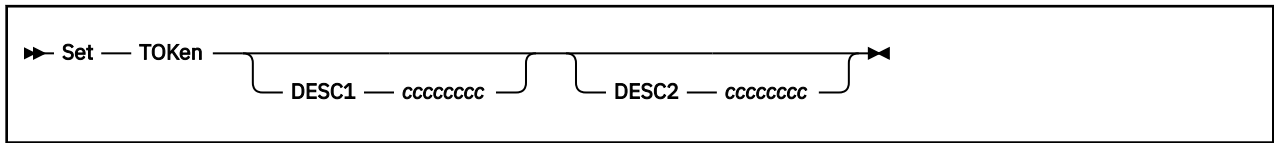
```
HCP6759I The time zone has changed to zone
```

CP sends the same message to each user who is logged on or disconnected from the system. Users can suppress the message by using the SET EMSG command. For more information on the SET EMSG command, see [“SET EMSG” on page 1512](#).

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP6704E Missing token at end of line
- HCP6706E Invalid time zone ID - *zone*
- HCP0987E SET TIMEZONE not valid - STP timezone in use

## SET TOKEN



### Authorization

Privilege Class: B, E

### Purpose

Use SET TOKEn to set the values of the channel subsystem configuration token description fields.

### Operands

#### DESC1 cccccccc

is the value you want assigned to the first description field of the channel subsystem configuration token. The variable *cccccccc* must be a 1- to 8-alphanumeric character string. If *cccccccc* contains blanks you must enclose the value in single quotation marks, for example, SET TOKEn DESC1 'name one'. If this operand is omitted, no change is made to the first description field.

#### DESC2 cccccccc

is the value you want assigned to the second field of the channel subsystem configuration token. The variable *cccccccc* must be a 1- to 8-alphanumeric character string. If *cccccccc* contains blanks you must enclose the value in single quotation marks, for example, SET TOKEn DESC2 'name two'. If this operand is omitted, no change is made to the second description field.

### Usage Notes

1. The SET TOKEn command is provided to allow the system programmer to set the description fields (DESC1 and DESC2) within the hardware I/O-configuration token. Setting these fields can be useful when you are running z/OS as either a guest on VM or in other LPARs on a system where VM is the controlling logical partition for dynamic-I/O configuration. z/OS uses the DESC1 field as the high-level dataset-name qualifier of the z/OS I/O-definition files (IODF). They use the DESC2 field to select which of several IODFs are to be used, with the DESC2 field taking on values in the form "IODFnn", where nn can have a value from 00 to 99.

By setting the DESC1 and DESC2 fields to the proper values for a z/OS IODF that is to be used by one or more z/OS guests, these guest operating systems will automatically select that IODF for their I/O configuration when the guests are IPLed.

VM itself makes no use of the DESC1 and DESC2 fields.

2. CP puts the machine in configuration mode to change the description fields and takes the machine out of configuration mode once the changes are made. Changing the token description fields does not affect the I/O configuration.
3. When issuing SET TOKEn, VM must not already be in configuration mode, but must be enabled for dynamic I/O configuration.
4. The Set TOKEn command changes the token for the current (i.e. the running) I/O configuration, but does not change the token that will be established from the IOCDs (I/O-configuration data set) at the next power-on reset (POR) of your machine. To make the same change to the active IOCDs (i.e. the IOCDs that will be used at the next POR), you must do the following:
  - a. Run the IOCP program, specifying all of the options you normally specify, including WRTxx (to write the newly-generated IOCDs into the machine).

- b. Issue the Set IOCDS\_active command to make the new IOCDS the "active" IOCDS (i.e. the IOCDS to be used at the next POR).

### Examples

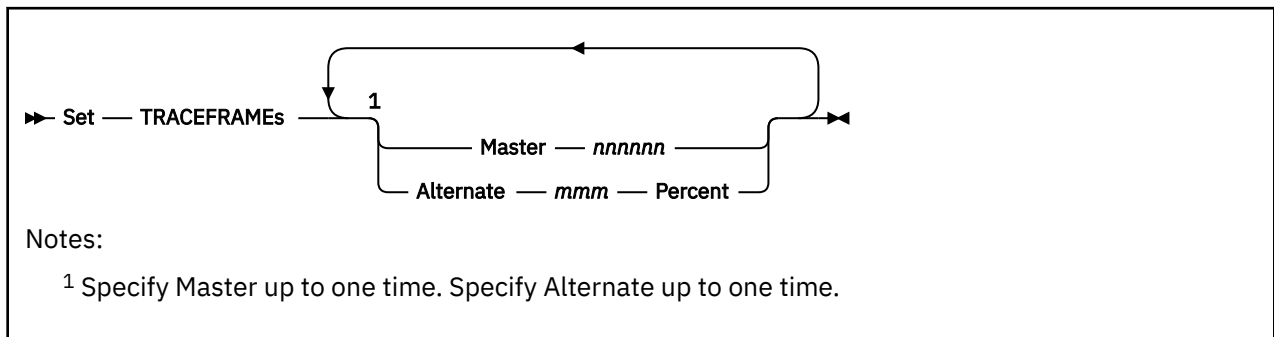
To change the first description field to 'IOCONFIG' and the second description field to 'IODF43', enter:

```
set token desc1 ioconfig desc2 iodf43
The channel subsystem configuration token was set as requested
Ready;
```

### Messages

- HCP6805E Cannot find RDEV for TYPE 1 subchannel
- HCP6806E Dynamic I/O change request failed with return code *nnnn*
- HCP6808E System disabled for dynamic I/O changes. The VM configuration token does not match the hardware configuration token.
- HCP6808E System disabled for dynamic I/O changes. The dynamic I/O support was disabled at the service console.
- HCP6809E This processor does not support dynamic I/O changes
- HCP6815E Command failed in the channel subsystem
- HCP6815E Cannot find configuration token in the system
- HCP6816E Dynamic I/O changes are not allowed on this system
- HCP6850E Cannot set system configuration token. A dynamic I/O change is in progress.
- HCP6850E Cannot set system configuration token. System is in configuration mode.
- HCP6851E I/O configuration cannot be changed because VM is not the controlling partition.

## SET TRACEFRAMES



### Authorization

Privilege Class: A

### Purpose

Use SET TRACEFRAMES to change the number of real storage frames that CP sets aside for an internal trace table for the master processor and the alternate processors in the configuration.

### Operands

#### Master *nnnnnn*

specifies the number of real storage frames for the master processor's internal trace table. The value *nnnnnn* must be equal to or greater than 3, and it cannot exceed 131072.

#### Alternate *mmm* Percent

specifies the size of each alternate processor's internal trace table, which is *mmm* percent of the master processor's internal trace table.

### Usage Notes

1. CP maintains separate internal trace tables for each processor in the configuration. You can use this command to set the size of the internal trace table for the master processor and the alternate processors, which are expressed as a percentage of the master processor's trace table.
2. Each increase in the size of the system internal trace tables should be small. You should break a large increase down into a number of smaller ones, each less than the size of the system's available frame queue. Use the QUERY FRAMES command to see how your system has allocated frames of real storage.
3. If CP processes a request to increase the internal trace table and all readily available storage has been allocated, CP returns the internal trace table to the size prior to the SET TRACEFRAMES command.
4. A TRSAVE command typically extracts data from the system internal trace table. A SET TRACEFRAMES command, therefore, will be rejected if a TRSAVE command is active. Similarly, a TRSAVE command will be rejected if a SET TRACEFRAMES command is active.
5. The number of trace frames for each processor will always be adjusted to a minimum of 3.
6. See QUERY TRACEFRAMES for information on how to display the size of the internal trace tables.
7. Trace pages associated with trace frames are in system execution space (SXS) and backed by a below-2-GB frame. This means that there is a limit on the number of trace frames due to the storage limit on SXS. An "insufficient storage" condition might occur when requesting more trace frames for each logical processor and will become more likely as the number of running logical processors in a configuration increases.

## Examples

If you issued the command:

```
SET TRACEFRAMES MASTER 100 ALTERNATE 90 PERCENT
```

and you had a four-processor CPU, the master processor would be allocated a 100-frame internal trace table, and each of the three alternate processors would be allocated a 90-frame internal trace table.

If you subsequently issued the command:

```
SET TRACEFRAMES MASTER 200
```

the master processor would be allocated a 200-frame internal trace table, and each of the three alternate processors would be allocated a 180-frame internal trace table.

## Responses

Response 1:

```
Command Complete
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *operand*
- HCP013E Conflicting option - *option*
- HCP6077E TRSAVE is active.
- HCP6704E Missing token at end of line
- HCP6706E Invalid number - *operand*
- HCP6740E Insufficient storage to complete this request

## SET UNDERSCORE



### Authorization

Privilege Class: G

### Purpose

Use SET UNDERSCORE to control whether an underscore character is used to separate 16-digit output values in CP command responses for your user ID.

### Operands

#### OFF

indicates that an underscore character will not be used to separate 16-digit output.

#### ON

indicates that an underscore character will be used to separate 16-digit output into two 8-digit entities for readability.

### Usage Notes

1. UNDERSCORE is set OFF when you log on and is not changed until you issue SET UNDERSCORE ON.
2. Use QUERY UNDERSCORE to determine the current setting.
3. CP commands affected by the UNDERSCORE setting include DEFINE STORAGE, DISPLAY (Registers), DUMP (Registers), LOCATE FRAMETBL, LOCATE (Storage), LOCATE SXSTE, LOCATE VMDBK, LOCATEVM, LOCK, QUERY SXSSTORAGE, TRACE, TRACE TABLE, and VMDUMP.
4. An underscore character is always allowed on input (for the commands listed above and other CP commands), regardless of the setting of the UNDERSCORE option. When specifying an underscore on input, there must be exactly eight characters following the underscore.

### Examples

If you specify SET UNDERSCORE ON, then enter a DUMP GG12 command, the output would be formatted as follows:

```
GRG 12 = 00000000_0105E2F8
```

If UNDERSCORE OFF is in effect when you enter the DUMP GG12 command, the output would be formatted as follows:

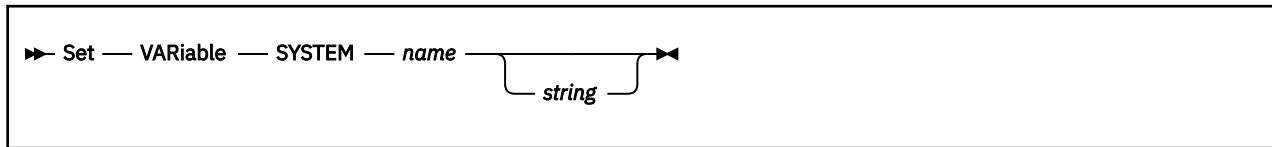
```
GRG 12 = 000000000105E2F8
```

### Messages

- HCP003E Invalid operand - *operand*
- HCP026E Operand missing or invalid



## SET VARIABLE



### Authorization

Privilege Class: B

### Purpose

Use SET VARIABLE to define and set an environment variable that is accessible to every class G user on the system. SET VARIABLE can also be used to change the value of an environment variable or to delete it by setting it to the null string. A *null string* is a string that contains no characters. Null strings have a length of zero.

System environment variables allow automated programs to adapt to different operating environments or modes in a coordinated fashion across multiple guest virtual machines.

### Operands

#### SYSTEM

indicates that this is a system environment variable.

#### name

specifies the name of the environment variable. This name must satisfy the following rules:

- It is 1 - 63 characters in length with no blanks.
- Valid characters are 0-9, A-Z, \_ (underscore), and . (period).
- It cannot begin or end with a period.
- It cannot begin with a "CP." prefix.
- It cannot begin with a numeric digit.

#### string

is the data string that will become the value for the specified variable. This string must satisfy the following rules:

- It is 1 - 255 characters in length.
- Valid characters are the hexadecimal values x'40' - x'FE'.
- Blanks (x'40') can be embedded within the string only. Blanks are stripped from the beginning and end of the string.

If the string is omitted, the specified variable is deleted.

### Usage Notes

1. For a complete list of environment variables defined, issue the QUERY VARIABLE ALL command.
2. When SET VARIABLE is specified for an environment variable that already has a value, the new value specified replaces the previous value.
3. Variable names that start with the prefix "CP." are reserved for IBM use and cannot be defined, changed, or deleted by the SET VARIABLE command.
4. A maximum number of 1000 system environment variables can exist at a time.
5. If SET VARIABLE is issued to delete a variable that does not exist, the command completes without error.

## SET VARIABLE

6. The maximum allowable length for a Diagnose x'08' command is 240. Subtracting 13 characters for:

```
"S VAR SYSTEM "
```

and one character for the blank following the variable name leaves a total of 226 characters to be used as the sum of the lengths of both the name and value.

7. If the value contains a character that is used as a line-editing symbol, the CP TERMINAL command should be issued first to define a different line-editing symbol or to turn off the line-editing function. Issue the CP QUERY TERMINAL command to view the characters that are used as line-editing symbols.

8. A string containing lowercase characters can be defined for the environment variable value only when the command is specified through the DIAGNOSE X'08' programming interface. This includes specifying the command within a REXX EXEC, which uses the Diagnose x'08' interface. When the command is specified on the command line, CMS or CP automatically translates the input data to uppercase so it is not possible to provide lowercase input from the command line.

Here is an example of using a REXX EXEC to specify a lowercase string that includes embedded blanks:

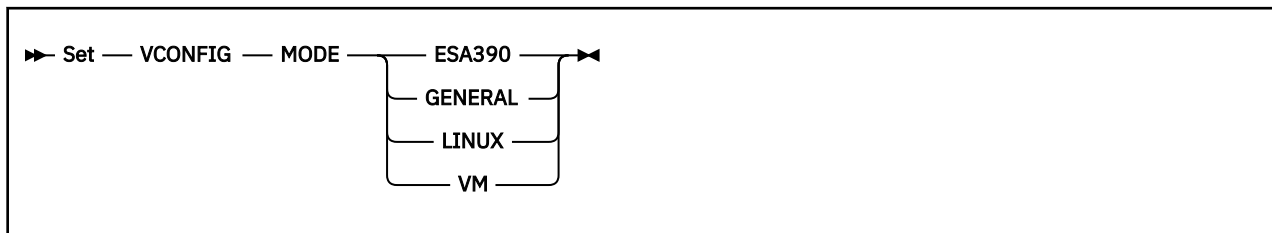
```
(a sample REXX exec)
/* This comment format indicates the REXX interpreter is to be used */
address command "CP SET VAR SYSTEM TESTLVL4 kag 002/b 12-03 2015"
```

Note that within the command string being executed, CP SET VAR must be uppercase.

### Messages

- HCP0002E Invalid operand - *operand*
- HCP1014E A required option is missing
- HCP2147E Environment variable is not set because the maximum number of environment variables are already defined.
- HCP2149E A variable name starting with reserved prefix "CP:" cannot be set.
- HCP2768E Missing variable name
- HCP6706E Invalid variable name - *name*

## SET VCONFIG



### Authorization

Privilege Class: G

### Purpose

Use SET VCONFIG to set the virtual configuration mode for a virtual machine.

### Operands

#### MODE

indicates that the configuration mode of the virtual machine is being changed.

#### ESA390

indicates that the configuration mode of the virtual machine is ESA/390, and only virtual CPU types that are valid in an ESA/390 mode virtual machine can be defined. VCONFIG modes ESA390 and GENERAL are functionally equivalent. If ESA390 is specified while MACHINE is set to Z, the system behaves as if GENERAL was specified. See the description of GENERAL for more information. ESA390 is the default setting for virtual machines at logon when the real logical partition mode is not Linux-only and the virtual machine mode is XA, ESA, or XC.

#### GENERAL

indicates that the configuration mode of the virtual machine is GENERAL. VCONFIG modes ESA390 and GENERAL are functionally equivalent, which means that the same virtual CPU types can be defined in an ESA390 mode virtual machine and in a GENERAL mode virtual machine. For the SET VCONFIG command, ESA390 and GENERAL are synonyms and cause a configuration of the same capabilities to be defined when specified. The SET VCONFIG command, like the QUERY VCONFIG command, will display ESA390 when MACHINE is set to XA, ESA, or XC, and will display GENERAL when MACHINE is set to Z. GENERAL is the default setting for virtual machines at logon when the real logical partition mode is not Linux-only and the virtual machine mode is Z.

#### LINUX

indicates that the configuration mode of the virtual machine is LINUX. This is the default setting for virtual machines at logon and the only valid configuration mode when the real logical partition mode is Linux-only.

#### VM

indicates that the configuration mode of the virtual machine is VM and only virtual CPU types that are valid in a VM mode virtual machine can be defined.

### Usage Notes

1. The virtual configuration mode settings correspond to logical partition modes which can be set on the General page of the image profile for a logical partition.
2. Changing the virtual configuration mode setting of a virtual machine results in a System Clear operation.

3. The virtual configuration mode determines the types of virtual CPUs that can be defined for this virtual machine. Refer to the description of “[DEFINE CPU](#)” on [page 152](#) to see the possible CPU types that can be defined for each virtual configuration mode.
4. To make changes to a virtual configuration, including changing virtual CPU types and the virtual configuration mode, do the following:
  - Issue the DEFINE CPU command to change all CPU types to CP.
  - Issue the SET VCONFIG MODE command to change the virtual configuration mode.
  - Issue the DEFINE CPU command to change the CPU types to the desired types.
5. While an implicit configuration mode of CF (Coupling Facility) exists, CF cannot be specified as a mode on the SET VCONFIG command. When OPTION CFVM is specified in the user directory entry, the virtual configuration mode is set automatically to CF and cannot be changed to a different mode by the SET VCONFIG command.

### Responses

Response 1:

```
MODE = m
```

where:

**m**

identifies the configuration mode and is set to one of the following:

```
ESA390  
GENERAL  
LINUX  
VM
```

Response 2:

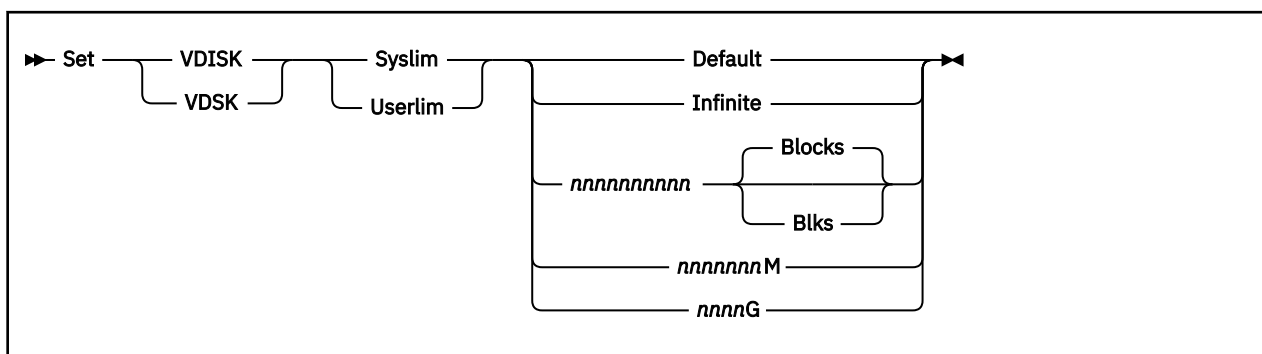
```
STORAGE CLEARED - SYSTEM RESET
```

is issued whenever the virtual configuration mode is changed to a different setting.

### Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *option*
- HCP1014E A required option is missing
- HCP1469E An attempt was made to define a virtual configuration mode that is not compatible with your virtual CPU configuration.
- HCP1470E The virtual configuration mode that was specified is not valid. It must be LINUX because CP is running in a Linux-Only logical partition.
- HCP1471E The virtual configuration mode that was specified is not valid. CF is the only valid virtual configuration mode for CFVM users.
- HCP5780E Adjunct does not support the *command subcommand* command

## SET VDISK



### Authorization

Privilege Class: B

### Purpose

Use SET VDISK to control the maximum amount of host real storage that is available for allocation as virtual disks in storage. Using virtual disks in storage increases the load on system paging, so you should set limits in proportion to the availability of paging space.

### Operands

#### Syslim

sets the total resource available for allocating virtual disks in storage on the system.

#### Userlim

sets the maximum resource available for virtual disks in storage created by a single user using the DEFINE command. This limit does not apply to virtual disks in storage defined by MDISK statements in the directory.

#### Default

sets the limit to the default value. If an installation default is defined in the system configuration file, that value is used. If an installation default is not defined in the system configuration file, or if the file is not found, the built-in default is used, as follows:

- The built-in default user limit is 0.
- The built-in default system limit is the minimum of:
  - The amount of virtual storage that can be represented by one-quarter of the usable dynamic paging area (DPA), based on the fact that each gigabyte of virtual disk defined requires 2050 frames of host real storage
  - The amount of storage represented by one-quarter of the paging space defined for CP use

#### Infinite

indicates all available host real storage may be allocated to virtual disks in storage up to the limit of 2147483648 512-byte blocks.

#### nnnnnnnnnn Blocks

#### nnnnnnnnnn Blks

specifies the number of 512-byte blocks of host real storage available for virtual disks in storage. If the number specified is equal to or greater than 2147483648, the limit is set to INFINITE.

#### nnnnnnnM

specifies the number of megabytes of host real storage available for virtual disks in storage. If the number specified is equal to or greater than 1048576, the limit is set to INFINITE.

### ***nnnnG***

specifies the number of gigabytes of host real storage available for virtual disks in storage. If the number specified is equal to or greater than 1024, the limit is set to INFINITE.

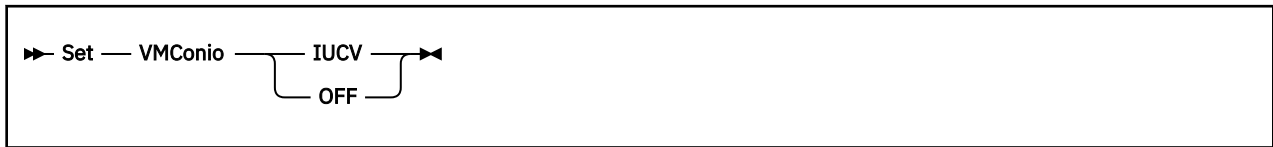
### **Usage Notes**

1. Virtual disks in storage are allocated in 8-block (4096-byte) pages. Therefore, the actual limit that is set may be rounded up to the nearest page.
2. If the system limit is set to a value that is less than the amount of host real storage currently allocated for virtual disks in storage, the command is accepted, but no new virtual disks in storage can be created until the total system usage drops below the limit. No currently-active virtual disks in storage are deleted. Any VDISK being created for a relocation in progress will not be subject to the new limit.
3. If the user limit is set to a value that is less than the amount of host real storage currently allocated for virtual disks in storage created by the DEFINE command by a given user, the command is accepted, but that user cannot create any new virtual disks in storage with the DEFINE command until the user's total usage of virtual disks in storage created with the DEFINE command falls below the limit. No currently-active virtual disks in storage are deleted. Any VDISK being created for a relocation in progress will not be subject to the new limit.
4. While virtual disks in storage can improve I/O performance, they may increase host real storage and paging resource requirements. Increasing the system and user limits increases the potential for end users to negatively impact system performance by using large virtual disks in storage improperly.
5. Regardless of possible USERLIM INFINITE specifications, the maximum number of USERLIM blocks is limited by the maximum number of blocks permitted by the command "CP DEFINE VFB-512". For more information, see ["DEFINE \(Virtual Disk in Storage\)"](#) on page 230.

### **Messages**

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - [command contains extra option(s) starting with] *option*
- HCP026E Operand missing or invalid

## SET VMCONIO



### Authorization

Privilege Class: G

### Purpose

Use SET VMCONIO to control the destination of the responses generated by your virtual machine. Responses include any line mode I/O between the virtual machine and the virtual machine console.

**Note:** When you log on, VMCONIO is set OFF.

### Operands

#### IUCV

specifies that responses are to be passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, data is handled as if SET VMCONIO OFF were issued.

#### OFF

specifies that responses are to be displayed at the terminal.

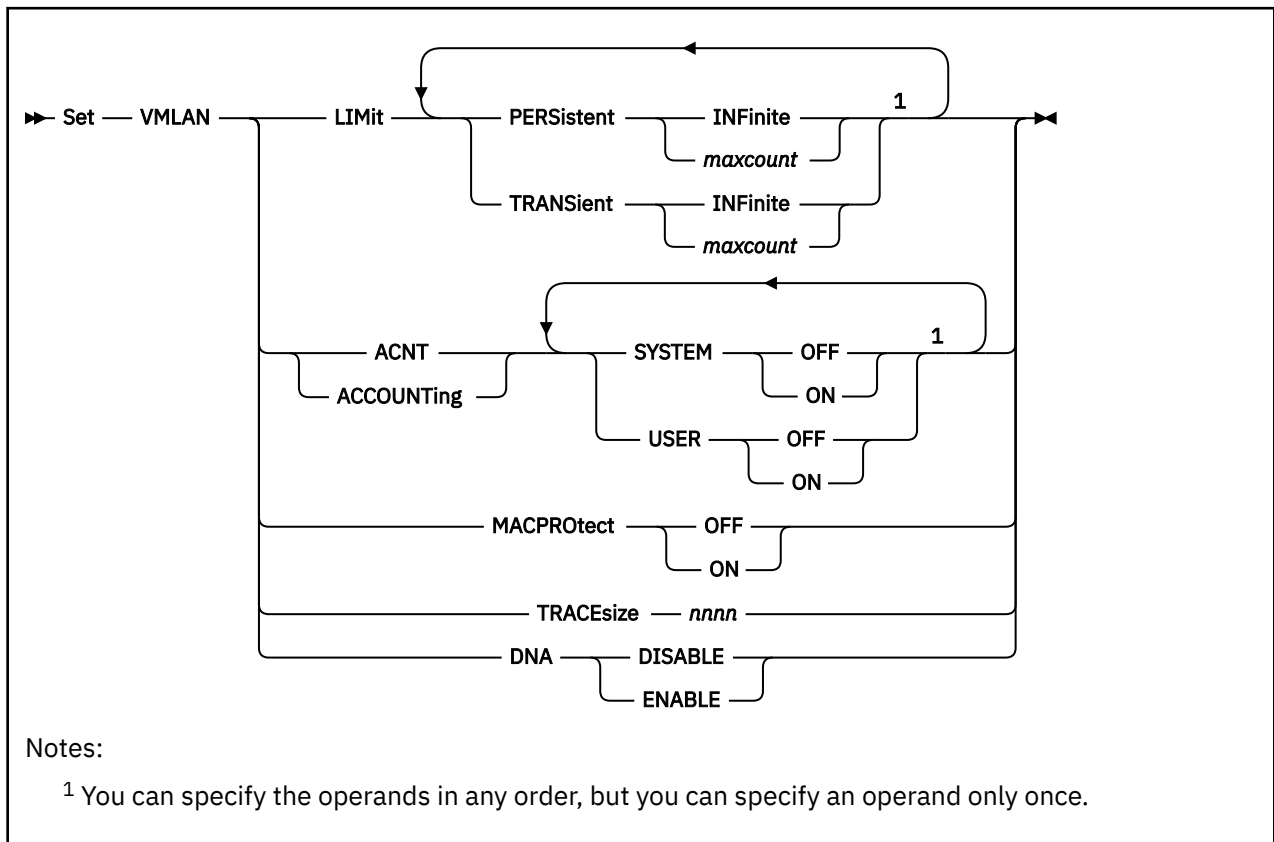
### Usage Notes

1. When full-screen CMS is on, most CMS console output is not passed to CP. In addition, applications that use the IUCV Message System Service (\*MSG) and SET VMCONIO IUCV will not trap all CMS output. Before running such applications, it is suggested that the user suspend full-screen CMS.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid

## SET VMLAN



### Authorization

Privilege Class: B

### Purpose

Use the SET VMLAN command to set system-wide attributes for virtual networking support.

### Operands

#### LIMit PERSistent INFinite|maxcount

specifies the maximum number of PERSISTENT guest LANs allowed on the z/VM system at one time. The value *maxcount* must be a decimal number in the range 0 - 1024. The default state of this attribute is INFINITE.

A persistent LAN is created when the DEFINE LAN system configuration statement is used to create a LAN during system initialization, or when a Class B user issues the DEFINE LAN command to define a SYSTEM owned LAN. A persistent virtual switch is created when the DEFINE VSWITCH system configuration statement is used to create the switch during system initialization, or when a Class B user issues the DEFINE VSWITCH command to define a virtual switch.

#### LIMit TRANSient INFinite|maxcount

specifies the maximum number of TRANSIENT guest LANs allowed on the z/VM system at one time. The value *maxcount* must be a decimal number in the range of 0 - 1024. The default state of this attribute is INFINITE.

A transient LAN is created when the DEFINE LAN command is used to define a general user LAN.



**ACNT|ACCOUNTING SYSTEM ON|OFF**

specifies the default accounting state for guest LANs owned by the SYSTEM user ID. The default state of this attribute is **OFF**. Class B users are authorized to override this setting on the DEFINE LAN and SET LAN commands.

**ACNT|ACCOUNTING USER ON|OFF**

specifies the default accounting state for guest LANs owned by individual users. The default state of this attribute is **OFF**. Class B users are authorized to override this setting on the DEFINE LAN and SET LAN commands.

**MACPROTECT**

turns MAC address protection on or off for virtual network devices on the system. Turning MAC address protection on prevents a guest from using a MAC address that is not assigned to the user's network device. MACPROTECT is valid only for ETHERNET virtual switches and guest LANs. MAC address protection is not supported for guest ports that are being bridged by this virtual switch.

There are three levels of inheritance used when determining the MAC address protection level for a MAC address assigned to a network device. The highest is the system level, followed by the virtual switch or guest LAN level. The lowest level is the protection set for a specific network data device.

The MAC address protection level assigned is determined first at the device, then virtual switch or guest LAN and lastly the system level. If MAC address protection is set at the device level using the SET NIC CP command, then its current setting is used. When the device is set to UNSPECIFIED, it will inherit the MACPROTECT setting of the virtual switch. When the virtual switch MACPROTECT setting is set to UNSPECIFIED, the SYSTEM's MAC address protection specified by this command will be used.

**OFF**

sets the MAC address protection level to off for all virtual switches, guest LANs and network devices whose MAC address protection setting are not specified (set to UNSPECIFIED). OFF is the default setting.

If MAC address protection is set to UNSPECIFIED by the SET NIC CP command for a specific device or a SET VSWITCH for virtual switch coupled devices or SET LAN for LAN coupled devices, then the device's MAC address protection will be set off.

**ON**

sets the MAC address protection level on for coupled devices defined with a MAC address protection set to UNSPECIFIED set the MAC address protection level to on for all virtual switches, guest LANs and network devices whose MAC address protection setting are not specified (set to UNSPECIFIED).

If MAC address protection is set to UNSPECIFIED by the SET NIC CP command for a specific device or a SET VSWITCH for virtual switch coupled devices or SET LAN for LAN coupled devices, then the device's MAC address protection will be set on.

**TRACEsize nnnn**

specifies the number of pages to be allocated for an internal trace table to keep track of general virtual networking events. The allowable range is 0 to 4095. The default value is 8 pages in order to increase the chance of capturing sufficient documentation in the event of a virtual networking problem. Increasing the page size allows a greater history of traces to be obtained when gathering documentation to assist IBM in diagnosing a virtual networking problem. Setting a value of 0 turns off this data capture function.

**DNA DISABLE|ENABLE**

Disable or Enable Directory Network Authorization

**DISABLE**

Turns off Directory Network Authorization management on the system. A NICDEF directory statement specified by the system administrator will not authorize the NIC defined to connect to the specified virtual network. Either a SET VSWITCH PORTNUM | GRANT CP command or a MODIFY VSWITCH System CONFIG statement must be issued on the target VSwitch to authorize the connection and to specify the NIC characteristics. Therefore, NICDEF PORTNUM, PORTTYPE, VLAN, PQUPLINKTX, PROMISCUOUS, and NOPROMISCUOUS operands will be ignored when VMLAN DNA DISABLE is specified.

### ENABLE

Turns on Directory Network Authorization management on the system (default setting). This will allow a system administrator to create and authorize a virtual NIC created by a NICDEF directory statement to connect to a virtual network. All NIC attributes and VSwitch authorization can be performed via the system directory instead of issuing either SET VSWITCH PORTNUM | GRANT CP commands or a MODIFY VSWITCH System CONFIG statement. Authorization is only granted automatically when the target VSwitch is not managed by an ESM.

### Usage Notes

1. Use the QUERY VMLAN command to find out the current attributes in effect for the running CP system.
2. The SET VMLAN statement can be used to set a limit that is lower than the current number of guest LANs. This will **not** force the elimination of any existing guest LANs. It will prevent the creation of new guest LANs until the current number drops below the stated limit.

### Examples

Set a limit of 20 TRANSIENT guest LANs:

```
set vmlan limit transient 20
command complete
```

Prevent the creation of any new TRANSIENT guest LANs:

```
et vmlan limit transient 0
command complete
```

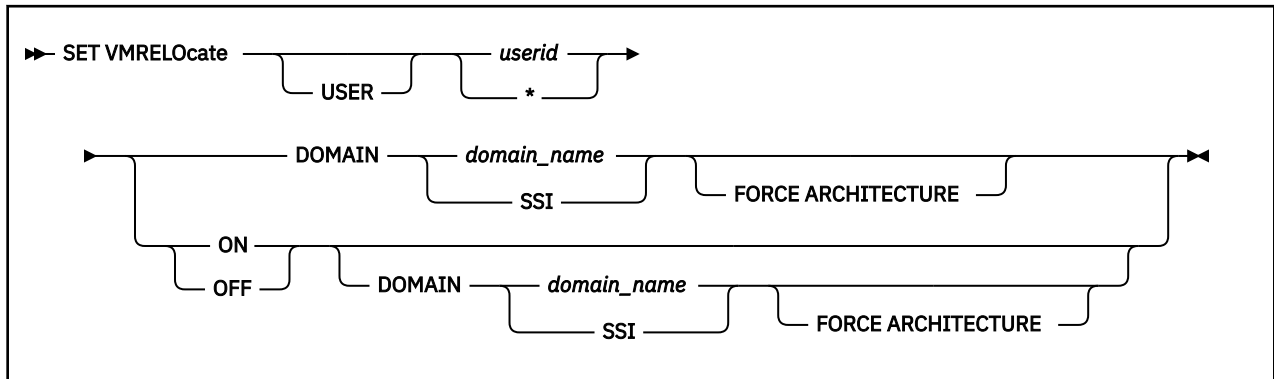
Enable accounting for every user-owned guest LAN:

```
set vmlan acct user on
command complete
```

### Messages

- HCP3195E SET [VMLAN | VSWITCH] not performed; trace table can not be allocated.
- HCP6645E SSI cluster in safe mode; command rejected.
- HCP6706E Invalid TRACESIZE value - *value*

## SET VMRELOCATE



### Authorization

Privilege Class: A

### Purpose

Use SET VMRELOCATE to dynamically control the relocation capability of a specified user. This command can set a user's relocation domain, indicating the set of systems to which it may be relocated, or can disallow relocation of this virtual machine. The virtual architecture level of the virtual machine is the maximal common subset of the architectural features of the cluster members in the relocation domain.

### Operands

#### *user*

is the user ID whose relocation capability is being set. If "\*" is specified, the target user is the command issuer.

A relocation domain cannot be set for a user ID defined by a multiconfiguration virtual machine definition.

#### **DOMAIN** *domain\_name*

defines the domain currently associated with a user. If DOMAIN is not specified, the current domain association is not changed. If a new domain is specified, the user is assigned the new virtual architecture level immediately. This could mean a downgrade in the guest's virtual architecture level. In that case, the FORCE ARCHITECTURE option must be specified. The *domain\_name* can be specified as SSI, which is the domain that includes all members of the SSI cluster.

#### **ON**

enables relocation for the specified user. If DOMAIN is not specified, the currently associated domain is assumed. If the user is not logged on, an error is returned. If neither ON nor OFF is specified, the relocation enablement is not altered.

#### **OFF**

disables relocation for the specified user. If DOMAIN is not specified, the currently associated domain is assumed. If the user is not logged on, an error is returned. If neither ON nor OFF is specified, the relocation enablement is not altered.

#### **FORCE ARCHITECTURE**

specifies that the virtual machine should be assigned to the new domain, even if it means the guest's virtual architecture will be set to a level with less capability than it had in its original domain.



**Attention:** Extreme care should be taken when using this option. If the unavailable hardware features or CP capabilities are important to the virtual machine or its applications, it is likely that the virtual machine or its applications will fail after the domain change is complete.

## SET VMRELOCATE

The guest's architecture is never changed if the SET VMRELOCATE command assigns the guest to the same domain to which it was already assigned, even if the FORCE ARCHITECTURE option was specified.

### Usage Notes

1. Use the VMRELOCATE directory statement rather than including a SET VMRELOCATE command in the directory entry of a user.
2. For information about the virtual architecture level and [Using Relocation Domains](#), see *z/VM: CP Planning and Administration*.

### Responses

Response 1:

```
Running on member sysname  
Relocation enabled|disabled in Domain domain
```

Response 2:

```
This system is not a member of an SSI cluster
```

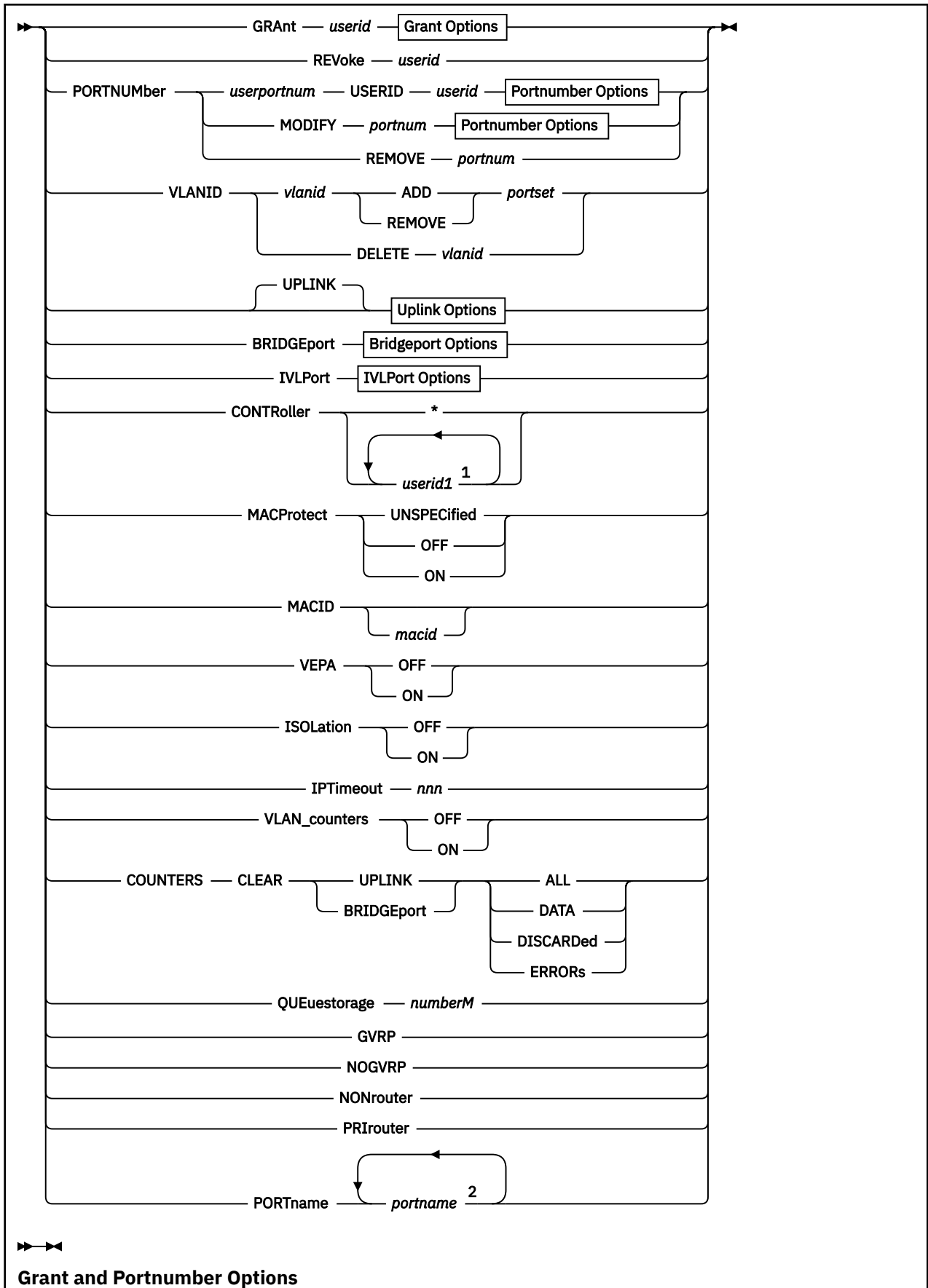
### Messages

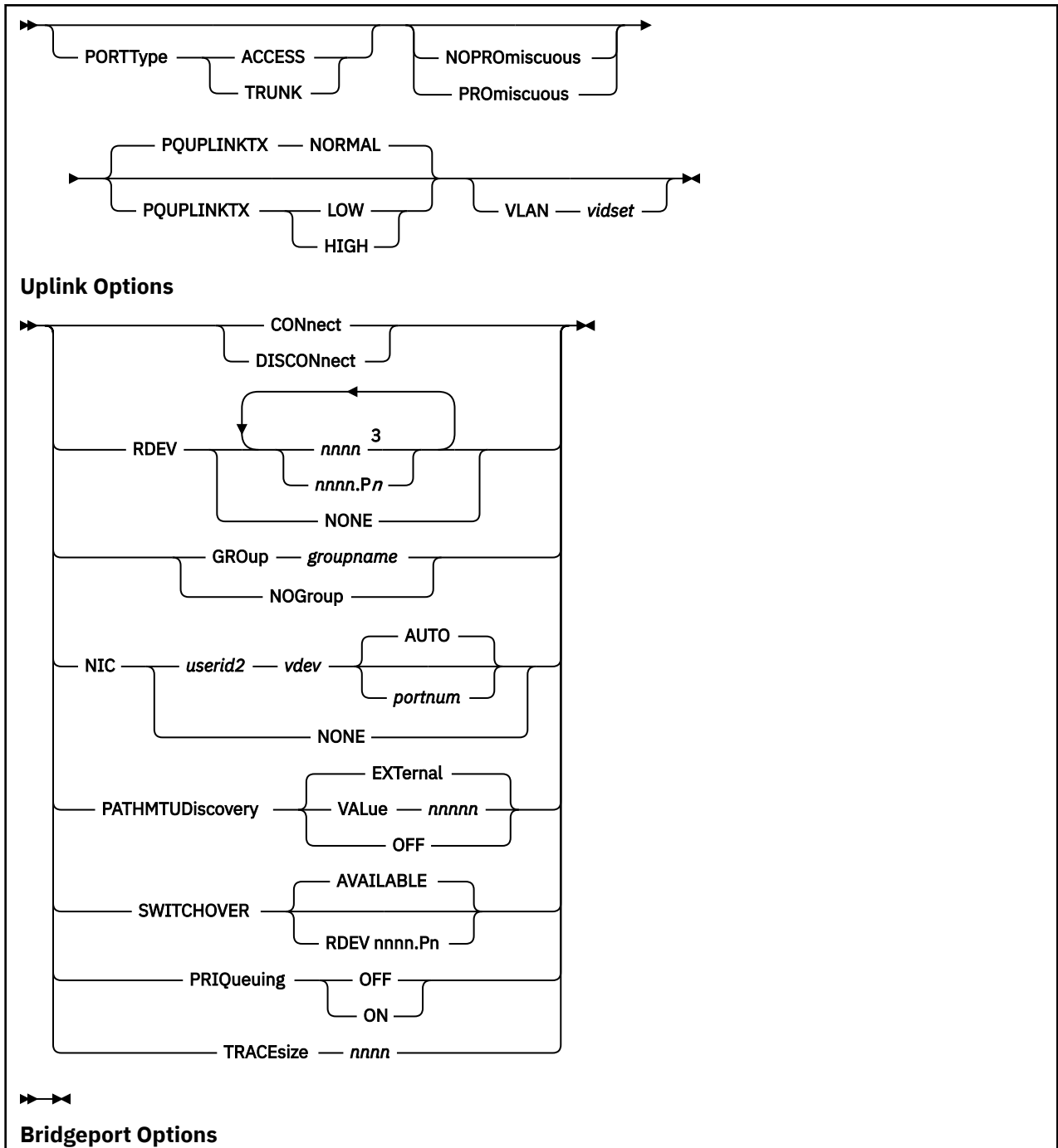
- HCP0002E Invalid operand - *operand*
- HCP0003E Invalid option - *option*
- HCP0013E Conflicting option - *option*
- HCP0026E Operand missing or invalid
- HCP0045E *userid* not logged on
- HCP1818I Excluded members: *mem1 mem2 mem3 mem4*
- HCP1819E The command could not be processed because the relocation domain specified is marked for deletion
- HCP1821E Relocation domain *domain\_name* does not exist
- HCP1822E User *userid* cannot be set to relocation domain *domain\_name* without the FORCE ARCHITECTURE option
- HCP1823E A multiconfiguration virtual machine cannot be relocated
- HCP1831E The command could not be completed because of a temporary condition. Try again later.
- HCP6706E Invalid string - *string*

## SET VSWITCH

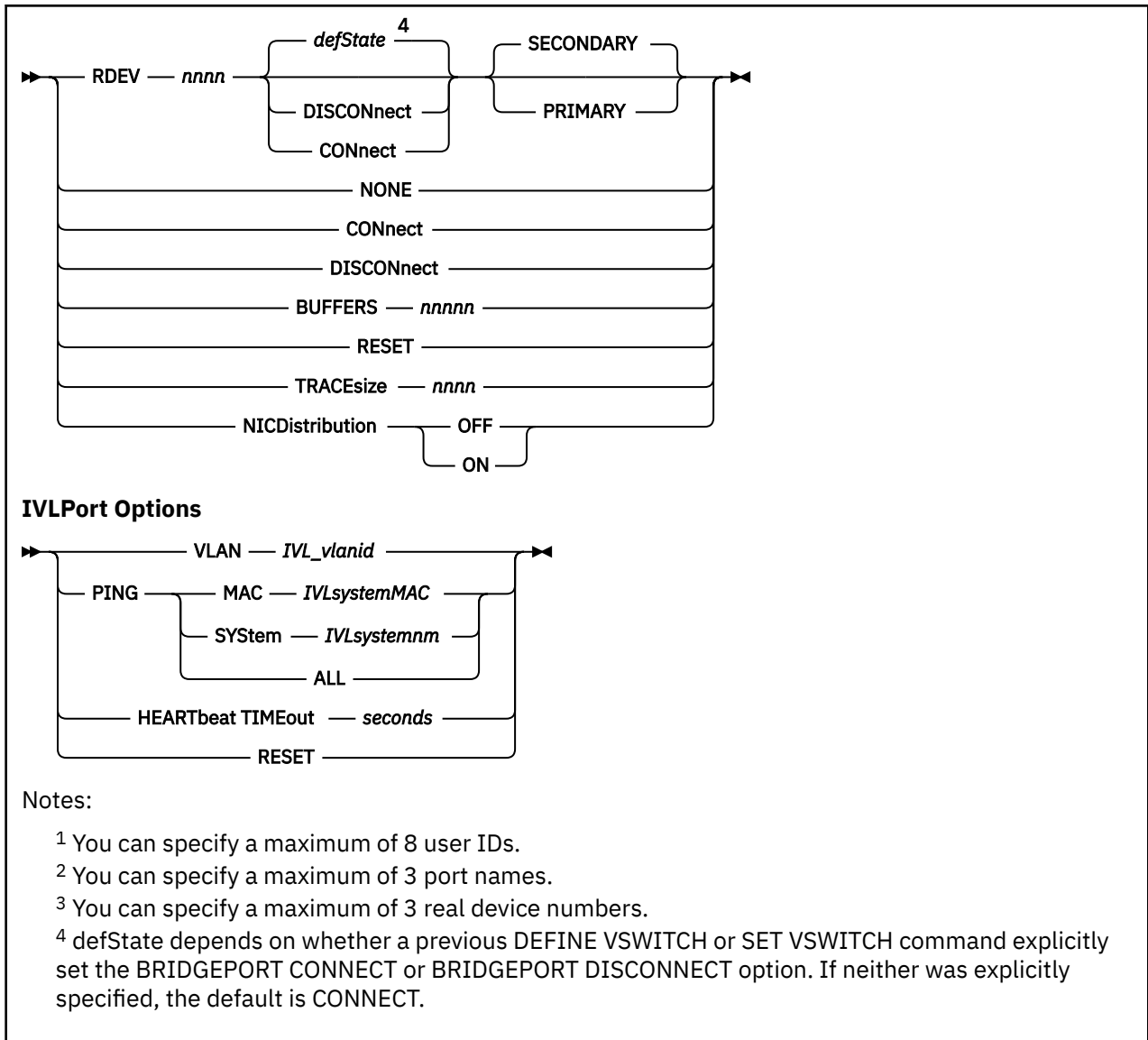
---

▶ Set VSWITCH — *switchname* ◀





## SET VSWITCH



### Authorization

Privilege Class: B

### Purpose

Use the SET VSWITCH command to set the attributes of an existing virtual switch.

### Operands

#### **switchname**

is the name of the virtual switch to be set. The *switchname* is a single token (1–8 alphanumeric characters) that identifies the virtual switch. This virtual switch was created by a DEFINE VSWITCH command or statement in the System Configuration file.

#### **GRANT userid**

specifies the *userid* to be either added or modified in the access list for this virtual switch. This allows *userid* to connect an adapter to *switchname*. If an External Security Manager (ESM) is in control of the virtual switch, it may override the CP access list.

If the *userid* is not currently in the access list, it will be added with the specified network characteristics. Any network characteristics not specified will use the defaults from the virtual switch.



If the *userid* already exists in the access list, then only the network characteristics specified on this command will be changed.

The SET VSWITCH command with GRANT creates a z/VM assigned port (in the range of 2176-4095 inclusive) representing the intended configuration if a port in this range does not already exist. The port defined by this command will persist until the VSWITCH is destroyed or the GRANT is removed by the SET VSWITCH command with the REVOKE option. If any ports are already configured for this user, update each port as specified by this new GRANT specification.

#### **PORTType**

defines the type of VSwitch connection to be used. If the GRANT or PORTNUMBER operand is being used to add a new user or port to the access list, and the virtual switch has been defined as VLAN aware and no PORTTYPE is specified on the GRANT or PORTNUMBER, the porttype assigned to the guest or port will be the default porttype as specified on the DEFINE VSWITCH command or statement.

If the GRANT or PORTNUMBER operand is issued for a user or port that has already been added to the access list, and the virtual switch has been defined as VLAN aware and no PORTTYPE is specified on the GRANT or PORTNUMBER, the current setting of porttype for the guest or port will be used. PORTTYPE TRUNK is not allowed for a TYPE IVL virtual switch.

#### **PORTType ACCESS**

defines the type of connections that are established to be an access port. The guest is unaware of VLAN IDs and sends and receives only untagged traffic.

#### **PORTType TRUNK**

defines the type of connections that are established to be a trunk port. The guest is VLAN aware and sends and receives only tagged traffic for those VLANs to which the guest is authorized. If the guest is also authorized for the *natvid*, untagged traffic sent or received by the guest is associated with the native VLAN ID of the virtual switch.

#### **VLAN vidset**

identifies the VLAN ID (or set of VLAN IDs) to which this user is restricted while connected to *switchname*. If VLAN is not specified, the default VLAN for this user is the default VLAN ID as specified on the DEFINE VSWITCH command or statement. Note that when a virtual switch is defined as VLAN AWARE, a default VLAN ID is not assigned. If a default VLAN ID is not assigned when the virtual switch is defined, then all inbound or outbound frames are discarded until a VLAN ID is assigned.

If VLAN is specified, each item in the *vidset* must be a valid VLAN number (between 1 and 4094, inclusive). The *vidset* may contain ranges:

*Table 39. vidset Ranges*

---

#### **Examples:**

---

VLAN 100-199

VLAN 100-199 1000-1099

---

VLAN 1 101-199

---

VLAN 1 101-199 4094

---

VLAN 1 2 3

---

VLAN 1 101 201 301

---

If an External Security Manager (ESM) is in control of the virtual switch, it can override the CP access list for a virtual switch.

#### **PROMiscuous**

specifies that the user ID or port is authorized for PROMISCUOUS Mode.

### **NOPROMiscuous**

specifies that the user ID or port is NOT authorized for PROMISCUOUS Mode. This is the default setting when the GRANT or PORTNUMBER operand is being used to add a new user or port to the access list.

If an External Security Manager (ESM) is in control of the virtual switch, it may override the PROMISCUOUS authority.

### **PQUPLINKTX**

specifies the priority for all packets sent from a virtual NIC's network connection to an external network. The PQUPLINKTX value will be used when priority queuing is enabled on the virtual switch (PRIQueuing ON). If PQUPLINKTX is configured for a virtual switch that does not have priority queuing enabled, the setting will be saved and used if priority queuing is enabled at a later time. If PQUPLINKTX is not specified, the default is that all outbound traffic to the external network will be sent at Normal priority. See PRIQueuing operand for additional information.

This option is not allowed for IVL virtual switches. For an IVL virtual switch, z/VM handles the priority of outbound transmissions.

Outbound traffic from the Hipersockets Bridgeport will be sent at Normal priority.

### **LOW**

specifies that outbound traffic to the external network will be sent at a low priority. This traffic will use the low priority queue which is serviced less frequently than the normal or high priority queues.

### **NORMAL**

specifies that outbound traffic to the external network will be sent at a normal priority. This traffic will use the normal priority queue which is serviced less frequently than the high priority queue but more frequently than the low priority queue.

### **HIGH**

specifies that outbound traffic to the external network will be sent at a high priority. This traffic will use the high priority queue which is serviced more frequently than the normal or low priority queues.

### **REVOKE *userid***

specifies *userid* is removed from the CP access list for this virtual switch. This prevents user *userid* from connecting an adapter to virtual switch *switchname*. If an External Security Manager (ESM) is in control of the virtual switch, it may override the CP access list.

The REVOKE option removes configured ports belonging to *userid* whether they were configured by SET VSWITCH with GRANT, PORTNUMBER, or NICDEF.

**Note:** If the connection was authorized by the ESM and it needs to be permanently removed, the ESM database must also be updated. If authorized by NICDEF, the user directory must also be updated (to permanently remove access).

### **PORTNUMBER *userportnum* USERID *userid***

The PORTNUMBER operand defines a port for a virtual switch and associates the specified user *userid* with a virtual switch port *userportnum*. This operand will also be used to specify port options including porttype, promiscuous, and VLAN ids. The *userid* will be added to the access list for this virtual switch. This allows *userid* to connect an adapter to *userportnum* on virtual switch *switchname*. If an External Security Manager (ESM) is in control of the virtual switch, the *userid* must be authorized by the ESM, and the VLANs configured with the SET VSWITCH PORTNUMBER command must be included in the VLAN list provided by the ESM. The valid range for user-defined ports is 1-2048.

This option is not allowed for a TYPE IVL virtual switch.

### **PORTNUMBER MODIFY *portnum***

This operand is used to change the port options including porttype, promiscuous, and VLAN ids. The valid range is 1-4095 because the administrator can modify user-defined ports or ports allocated by z/VM.

This option is not allowed for a TYPE IVL virtual switch.

**PORTNUMBER REMOVE *portnum***

specifies the *portnum* to be removed from the virtual switch. Connections to this port are not allowed unless the port is again added to the virtual switch. Existing connections to this port on the virtual switch will be uncoupled. If the *userid* associated with this port does not have any other ports defined on this virtual switch, this will also revoke virtual switch access for the *userid*. The valid range is 1-4095 because the administrator can remove user-defined ports or ports allocated by z/VM.

This option is not allowed for a TYPE IVL virtual switch.

**VLANID *vlanid* ADD *portset***

The ADD operand adds the port set to the specified virtual lan segment. The *vlanid* must be a valid VLAN number (between 1 and 4094, inclusive). Each item in the *portset* must be a valid port number (between 1 and 4095, inclusive). Each port must be defined with SET VSWITCH PORTNUMBER, SET VSWITCH GRANT, or NICDEF before being added to a VLAN. The *portset* may contain ranges:

Table 40. *portset* Ranges

---

**Examples:**


---

1-25

---

1 25-50

---

1 25-50 2048

---

1 2 3

---

1 101 201 301

---

Existing connections will be updated.

This option is not allowed for a TYPE IVL virtual switch.

If an External Security Manager (ESM) is in control of the virtual switch, the VLAN ids defined with VLANID command must be included in the VLAN list provided by the ESM.

**VLANID *vlanid* REMOVE *portset***

The REMOVE operand removes the ports listed from the specified virtual lan segment. The *vlanid* must be a valid VLAN number (between 1 and 4094, inclusive). Each item in the *portset* must be a valid port number (between 1 and 4095, inclusive). The *portset* may contain ranges. See [Table 40 on page 1763](#). Existing connections will be updated. When the last port number is removed from the VLAN, the VLAN is deleted.

This option is not allowed for a TYPE IVL virtual switch.

**VLANID DELETE *vlanid***

The DELETE operand deletes the VLAN *vlanid* from the specified virtual switch. The *vlanid* must be a valid VLAN number (between 1 and 4094, inclusive). The VLAN will be removed from all ports. Existing connections will be updated.

This option is not allowed for a TYPE IVL virtual switch.

**UPLINK**

enables and specifies connectivity for a virtual switch UPLINK port. The UPLINK port is a special port that typically is used to connect the virtual switch to a physical switch -- essentially bridging the virtual switch's simulated network to a physical network.

**CONNECT**

indicates that the currently configured virtual switch UPLINK port must be activated, and traffic may flow through the devices associated with the specified UPLINK ports device(s). For the case when a virtual switch has been defined with the NOUPLINK operand, the CONNECT is not performed. If an error occurs with one of the backup devices, issue the SET VSWITCH *switchname* CONNECT command to refresh the backup device once the problem is corrected.

**DISCONnect**

indicates that the currently configured virtual switch UPLINK port must not be activated, and that no traffic is to flow through the devices associated with the specified UPLINK ports device(s).

A virtual switch can be functional without a connection to a real LAN segment, and traffic flows only between virtual machines coupled to the virtual switch.

**RDEV *nnnn*****RDEV *nnnn.Pn***

is a real device number or a real device number and OSA-Express port number to be used as an UPLINK to connect the virtual switch to a OSA-Express device. The device selected must be compatible to the type of virtual switch created. Specify each real device number as a hexadecimal number between X'0001' and X'FFFD'. If specifying a real device and an OSA-Express port number, specify the real device number as a hexadecimal number between X'0001' and X'FFFD' followed by a period (.), the letter 'P' or ('p') followed by the port number as a hexadecimal number between X'0' and X'F'. For example, to specify port 1 for RDEV 300, specify "300.P1". The value of the port number depends on how many ports the OSA-Express hardware adapter supports. If the port number is not specified, it will default to port 0.

You can specify a maximum of three real device numbers. If you specify more than one device number, each must be separated from the others by at least one blank. When the virtual switch has been defined with the GROUP attribute, any devices identified by the RDEV keyword are used for failover in the event of a real switch failure of the link aggregation group. Failover in this environment will be to a single OSA-Express device connected to a second real switch.

Each real device number represents a trio of devices. For example, specifying **RDEV 111 222 333** means that the first devices, 111-113, are used to provide the connection to the real hardware LAN segment. When issuing the VSWITCH command without the GROUP attribute, if there is a problem with the connection, devices 222-224 are used next to provide the connection, and if those devices fail to connect, devices 333-335 are used. This feature provides dynamic recovery for OSA-Express device failures when issuing the VSWITCH command without the GROUP attribute or for real switch failure when issuing the VSWITCH command with the GROUP attribute. (Failure of an OSA in an aggregated group is automatically handled by the virtual switch; that is, the virtual switch will transfer the data flow to the remaining OSAs in the group.)

RDEV NONE means that the virtual switch should not be connected to the real LAN segment when defined with NOGROUP. When the virtual switch has been defined with GROUP, RDEV NONE means that there is no link aggregation group failover in the event the real switch should fail.

The number of devices identified by the RDEV keyword combined with the number of devices in an associated GROUP may not exceed eight devices.

The RDEV operand may not be used to specify device numbers when the virtual switch is configured to use a shared port group. MAC address takeover is managed by the shared port group to maintain connectivity following a failure.

**GRoup *groupname*****NOGroup**

GROUP indicates that the virtual switch is to be configured to use IEEE 802.3ad Link Aggregation. The *groupname* is a single token (1-8 alphanumeric characters) that identifies the group. Use the SET PORT GROUP command to specify the attributes of the group and the OSA-Express devices that will comprise the group. This option can be specified only when the virtual switch has been defined with the ETHERNET transport attribute. NOGROUP means that the virtual switch will not use Link Aggregation.

The GROUP or NOGROUP operands can be used only when the virtual switch is in the DISCONNECT state. If a group already has been associated with a virtual switch and you wish to change to a different group, first issue SET VSWITCH with the NOGROUP operand, followed by SET VSWITCH with a GROUP operand that specifies the new *groupname*. The port group must be defined using the SET PORT GROUP command before it can be associated with a GLOBAL virtual switch. For a LOCAL virtual switch, if the group specified has not been defined, it will be created with this command.

**NIC**

causes the UPLINK port for a virtual switch to be connected to, or disconnect from, a virtual switch guest port. The NIC operand can be specified only when a virtual switch is in the DISCONNECT state. Setting the UPLINK port to a guest port also will set RDEV NONE and NOGROUP. Changing either of the RDEV or GROUP settings causes UPLINK NIC NONE to be set.

This option is not allowed for a TYPE IVL virtual switch.

***userid2 vdev (AUTO/portnum)***

specifies that the UPLINK port is to be connected to a guest port, and that the virtual switch is not to be connected to a physical switch. In this configuration the virtual switch routes all broadcasts and destination MAC or IP Address unicast frames that cannot be resolved within the simulated LAN segment to the specified guest port. If your virtual switch is VLAN AWARE, the guest port should be configured as a TRUNK port.

- AUTO indicates that the CP determines the virtual switch port number to be used.

If a single user port was defined for this user by the SET VSWITCH command, then that port will be automatically used by COUPLE or CONNECT, and its port settings will be used. If there are multiple user ports defined, then the CONNECT or COUPLE command will fail.

- *portnum*

The guest portnum may be specified. When the uplink NIC is COUPLED or CONNECTED, it is connected to the specified *portnum* on the virtual switch. The portnumber must be a valid user-defined port number (between 1 and 2048, inclusive), and must be configured with the SET VSWITCH PORTNUMBER command. If the *portnum* is not a configured port, then the COUPLE or CONNECT command will fail.

**NONE**

disconnects the currently connected guest port from the special virtual switch UPLINK port. Until another UPLINK port is specified, all destination MAC or IP address unicast frames that cannot be resolved within the simulated LAN segment are discarded

**PATHMTUDiscovery**

indicates that CP will, by default, participate in the Path MTU Discovery process in order to minimize packet loss and to maximize the size of packets sent through a defined UPLINK RDEV port device. The Path MTU Discovery process performed by CP examines and discards any IP datagrams destined for the UPLINK port that are larger than the Path MTU Discovery value allowed by the UPLINK RDEV port device. CP also sends to the datagram source an appropriate notification in the form of an ICMP or ICMPv6 message.

The SET VSWITCH PATHMTUDISCOVERY option can be used to configure a user-defined Path MTU Discovery value, to indicate that CP should determine the Path MTU Discovery value (the default), or to indicate to CP that the Path MTU Discovery process is to be disabled.

These options are specified as follows:

**EXternal**

indicates that CP should determine the Path MTU Discovery value to assign. The value to be assigned is determined as follows:

- If the virtual switch is configured with an active UPLINK RDEV port, CP determines the Path MTU Discovery value to use from the maximum MTU value supported by the connected OSA UPLINK RDEV port device. Typically, this value is 8992 bytes.
- If the virtual switch is configured with one or more UPLINK RDEV ports but none are currently active, CP disables the Path MTU Discovery process, effectively the same as specifying the SET VSWITCH PATHMTUDISCOVERY OFF option. Later, when one of the UPLINK RDEV port devices is activated, CP determines the Path MTU Discovery value to use based on the maximum MTU value supported by the active OSA UPLINK RDEV port device. Typically, this value is 8992 bytes.
- If the virtual switch is configured without an UPLINK RDEV port device, CP disables the Path MTU Discovery process, effectively the same as specifying the SET VSWITCH

## SET VSWITCH

PATHMTUDISCOVERY OFF option. Later, when the UPLINK RDEV port device is defined and activated for the virtual switch, CP determines the Path MTU Discovery value to use based on the maximum MTU value supported by the active OSA UPLINK RDEV port device. Typically, this value is 8992 bytes.

PATHMTUDISCOVERY EXTERNAL is the default setting for the virtual switch.

### **VALue *nnnnn***

configures a user-defined value of *nnnnn* bytes to be used by CP as the Path MTU Discovery value. Specifying this option can be useful if the PATHMTUDISCOVERY value of the physical network needs to be different from the PATHMTUDISCOVERY value used by CP when SET VSWITCH PATHMTUDISCOVERY EXTERNAL is specified. To determine the recommended PATHMTUDISCOVERY value, refer to the Maximum Transmission Unit (MTU) information in the hardware documentation associated with the OSA UPLINK RDEV port device. *nnnnn* is a number from 512 to 65535.

### **OFF**

indicates that the Path MTU Discovery process for this virtual switch is to be disabled by CP. This setting is not recommended when the virtual switch is defined with a Bridge Port.

### **SWITCHOVER**

initiate a controlled port change or failover from the currently active OSA port to another configured backup port with minimal network disruption. Although the network connection is quickly terminated, it is still possible that data in transit may be lost, causing some additional network recovery actions to occur.

This command is not valid for a link aggregation or GROUP configured uplink port. To remove a port from a link aggregation group, use the SET PORT GROUP LEAVE command.

### **AVAILABLE**

Switch to one of the available configured backup uplink devices configured on the virtual switch. The first available port in which a new network connection can be established will become the active uplink port. If none of the configured backup ports can establish an active network connection, then a network connection will be established back on the original device.

### **RDEV *nnnn.Pn***

Attempt to switch to the specified configured backup device. The RDEV and OSA-Express port number specified must already be configured as a backup device with the SET VSWITCH UPLINK RDEV command. Specify the real device number as a hexadecimal number between X'0001' and X'FFFF' followed by a period (.), the letter 'P' or ('p') followed by the port number as a hexadecimal number between X'0' and X'F'. When a port number is not specified using (.Pn), then port number zero will be used.

If for any reason a network connection cannot be established on the specified RDEV, the network connection will be reestablished backed on the original uplink port.

### **PRIQueuing**

enables or disables guest priority queuing support on all outbound data transmissions from the virtual switch uplink port to an external network. Priority queuing is a capability of the OSA-Express feature where multiple output queues are defined for a single network connection with each queue weighed by a priority on how often it gets serviced by the OSA-Express feature. The highest priority queue is serviced first and more often followed by the next highest priority and so on. No queue is starved and all will get serviced at some point by the OSA-Express feature. See [Virtual Switch Priority Queuing Function](#) in *z/VM: Connectivity* for more information on virtual switch exploitation of priority queuing.

If PRIQueuing is to be enabled on a virtual switch, then the OSA-Express features configured to the virtual switch's uplink port must be configured by IOCP to enable the feature within the adapter (PQ\_ON). (See "[DEFINE CHPID / PATH](#)" on page 132 for more information.

### **OFF**

The virtual switch will not exploit priority queuing. A single input queue is established for inbound transmissions from the external network and a single output queue is established for outbound

transmissions. All outbound data to the external network is transmitted with equal priority. This is the default for TYPE QDIO virtual switches.

The OFF option is not allowed for a TYPE IVL virtual switch.

## ON

The virtual switch will exploit priority queuing. If the customer configured the OSA-Express features used by a virtual switch uplink port for priority queuing via IOCP (PQ\_ON), then z/VM will establish one input queue and four output queues when activating its network connection. This will allow z/VM to transmit data to the external network at four different priorities. CP will use the highest priority queue for control and management traffic. The other three queues (low, normal and high) can be used for virtual NIC's network connections. This is the default for TYPE IVL virtual switches.

For an IVL virtual switch, z/VM will attempt to establish an active network connection with the first OSA device specified. For migration reasons, if the OSA-Express features (used by the IVL virtual switch uplink port) are configured without priority queuing (PQ\_OFF), z/VM will establish an active network connection, and force priority queuing off. A warning message will be displayed to inform the customer to configure priority queuing via IOCP.

If the OSA-Express features used by a non-IVL virtual switch uplink port are configured without priority queuing (PQ\_OFF), z/VM will NOT establish an active network connection, and an error message will be displayed.

The policy used to select which priority a specific datagram is transmitted to the external network is determined by the type of virtual switch.

For an IVL virtual switch, the priority of outbound transmissions is handled by z/VM. IVL management traffic will be queued and transmitted on a high priority queue, and encapsulated production data failover traffic will be sent on a lower priority queue. The default for TYPE IVL virtual switches is PRIQUEUING ON, given priority queuing of both management and encapsulated production data is required to maintain a viable IVL network. Only an OSA-Express feature configured for priority queuing should be used on an IVL virtual switch's uplink port.

For all other virtual switch types, the priority can be set to low, normal or high for all packets sent from a guest NIC's network connection to an external network via the SET VSWITCH command. See the PQUPLINKTX operand on [“SET VSWITCH” on page 1757](#) for more information.

The PRIQueuing operand can be specified only when virtual switch's uplink port is in the DISCONNECT state.

## TRACEsize *nnnn*

specifies the number of pages to be allocated for an internal trace table to keep track of trace events pertaining to this UPLINK port. The allowable range is 0 to 4095. The default value is 8 pages in order to increase the chance of capturing sufficient documentation in the event of a virtual switch problem. Increasing the page size allows a greater history of traces to be obtained when gathering documentation to assist IBM in diagnosing an UPLINK related problem. Setting a value of 0 turns off this data capture function.

## BRIDGEport

enables and specifies connectivity for a virtual switch Bridge Port. The Bridge Port is a special port that is used to connect the virtual switch to a HiperSockets CHPID, essentially bridging the HiperSockets CHPID to the virtual switch's simulated and physical networks.

Configuring a virtual switch Bridge Port is only supported when the Bridge facility is supported by the processor, the virtual switch transport type is ETHERNET, the virtual switch TYPE is QDIO and the ISOLATION and VEPA modes are OFF.

Only guest operating systems running in a virtual machine under z/VM and exploiting QDIO Enhanced Buffer State Management (QEBSM) are eligible to be bridged from the HiperSockets CHPID to the virtual switch's simulated and physical networks.

**RDEV *nnnn***

is a real device to be used as a Bridge Port to connect the virtual switch to a HiperSockets CHPID. Only devices that have been configured with the HiperSockets CHPID parameters EXTERNAL\_BRIDGED, can be specified. The RDEV operand can be specified only when the Bridge Port is in the DISCONNECT state.

The device selected must be compatible to the type of virtual switch created; that is, EXTERNAL\_BRIDGED for a QDIO type virtual switch. Connectivity to the HiperSockets CHPID will be prevented when the specified device does not match the virtual switch's type.

Specify a single real device number as a hexadecimal number between X'0001' and X'FFFD'. The real device number specified represents a trio of devices. For example, specifying BRIDGEPORT RDEV 508 means the devices 508-50A are used to provide the connection to the HiperSockets CHPID.

**NONE**

specifies that the virtual switch does not have a Bridge Port. If a Bridge Port was previously defined, it must be in the DISCONNECT state in order to be removed.

**CONnect**

indicates that the device identified by the RDEV keyword will be immediately activated, allowing the connection to be used as the active or standby Bridge Port.

**DISCONnect**

indicates that the device identified by the RDEV keyword will be placed in the inactive state. If this connection is the active Bridge Port connection, another virtual switch with a Bridge Port on the same CHPID in standby state will take over the active Bridge Port connection.

**SECONDARY**

indicates that this virtual switch should be assigned a role as a SECONDARY Bridge Port for the HiperSockets CHPID. The SECONDARY operand can be specified only when the Bridge Port is in the DISCONNECT state. See the DEFINE VSWITCH usage note [“17” on page 242](#) for more information about the SECONDARY Bridge Port role.

**PRIMARY**

indicates that this virtual switch must be assigned the role as the PRIMARY Bridge Port for the HiperSockets CHPID. The PRIMARY operand can be specified only when the Bridge Port is in the DISCONNECT state. See the DEFINE VSWITCH usage note [“17” on page 242](#) for more information about the PRIMARY Bridge Port role.

**BUFFERS *nnnnn***

indicates the maximum number of asynchronous requests that are allowed to be outstanding on the Bridge Port. The allowable range is 0 to 65535. The default value is 2048.

**RESET**

is intended for diagnostic purposes when directed by IBM to synchronize the virtual switch and HiperSockets bridge communications. This RESET operand can be specified only when the Bridge Port is in CONNECT state and is the active Bridge Port connection.

**TRACEsize *nnnn***

specifies the number of pages to be allocated for an internal trace table to keep track of trace events pertaining to this Bridge Port. The allowable range is 0 to 4095. The default value is 8 pages in order to increase the chance of capturing sufficient documentation in the event of a virtual switch problem. Increasing the page size allows a greater history of traces to be obtained when gathering documentation to assist IBM in diagnosing a Bridge Port related problem. Setting a value of 0 turns off this data capture function.

**NICDistribution**

specifies how Ethernet frames flow from HiperSockets bridge capable ports on the HiperSockets CHPID into the virtual switch uplink or other simulated guests that are coupled to the virtual switch.

**OFF**

The virtual switch does not examine each Ethernet frame to identify the HiperSockets guest port that sent the frame. The virtual switch distributes all Ethernet frames from the



HiperSockets CHPID as a single source. OFF is the default setting if you do not specify the NICDISTRIBUTION parameter.

**ON**

The virtual switch examines each Ethernet frame to identify the HiperSockets guest port that sent the frame. The virtual switch uses the HiperSockets guest port information to distribute the frames among the virtual switch uplinks or other simulated guests that are coupled to the virtual switch. The information is used to balance the load, especially when links are in a link-aggregation port group configuration and when most guests are on the HiperSockets CHPID.

When NICDISTRIBUTION is ON, additional information that is related to the HiperSockets guest ports is displayed by the QUERY VSWITCH command, in monitor records, and by Diagnose x'26C'. For example, packet counters and IP addresses of the guests are displayed. For more information, see [QUERY VSWITCH](#) in *z/VM: CP Commands and Utilities Reference*.

**CONTROLLER \* | userid1**

identifies the z/VMz/VM user ID that controls the OSA-Express device connected to the virtual switch. You can specify a maximum of eight user IDs. If you specify more than one user ID, each must be separated from the others by a least one blank. CONTROLLER \* means CP selects from any of the eligible z/VM TCP/IP stacks. See usage note [“1”](#) on page 1773 for more information about the function of a controller.

If you specify multiple real devices on the RDEV keyword or through the GROUP keyword, then specify CONTROLLER \* or allow it to default or specify a list of user IDs. The controller functions are then spread across multiple z/VM TCP/IP stacks, providing more flexibility in case of a failure. You can also specify a pool of specific controllers to be chosen from by specifying a list of user IDs after the CONTROLLER keyword with the SET VSWITCH command or the MODIFY VSWITCH statement.

**MACProtect**

turn on or off MAC address protection for devices coupled to this virtual switch. Turning MAC address protection on prevents a guest from using a MAC address that is not assigned to the user's network device. MACPROTECT is valid only for an ETHERNET virtual switch. MAC address protection is not supported for guest ports that are being bridged by this virtual switch.

There are three levels of inheritance used when determining the MAC address protection level for a MAC address assigned to a network device. The highest is the system level, followed by the virtual switch (or guest LAN) level. The lowest level is the protection set for a specific network data device.

The MAC address protection level assigned is determined first at the device, then virtual switch and lastly the system level. If MAC address protection is set at the device level using the SET NIC CP Command, then its current setting is used. When the device is set to UNSPECIFIED, it will inherit the MACPROTECT setting of the virtual switch. When the virtual switch MACPROTECT setting is set to UNSPECIFIED, the SYSTEM's MAC address protection specified by the SET VMLAN CP command or VMLAN configuration statement is assigned.

**UNSPECified**

allow the MAC address protection to be determined by the SYSTEM level setting. UNSPECIFIED is the default setting for a virtual switch.

If MAC address protection is set to UNSPECIFIED by the SET NIC CP command for a specific device, then the device's MAC address protection will be set to the SYSTEM level defined by the SET VMLAN CP command or the VMLAN configuration statement.

**OFF**

set the MAC address protection level off for coupled devices defined with a MAC address protection set to UNSPECIFIED.

If MAC address protection is set to UNSPECIFIED by the SET NIC CP command for a specific device, then the device's MAC address protection will be set off.

**ON**

set the MAC address protection level on for coupled devices defined with a MAC address protection set to UNSPECIFIED.

## SET VSWITCH

If MAC address protection is set to UNSPECIFIED by the SET NIC CP command for a specific device, then the device's MAC address protection will be set on.

### **MACID** *macid*

is a unique identifier (up to six hexadecimal digits) used as part of the virtual switch MAC address. This MACID (three bytes) is appended to the system MACPREFIX (three bytes) or USERPREFIX (three bytes) to form a unique MAC address for this virtual switch. If a specified MACID is already in use, the virtual switch is not changed. You must eliminate the conflict or select a different MACID. If no MACID is set for the virtual switch, CP generates a unique identifier for this virtual switch. Also, specifying SET VSWITCH MACID without *macid* changes the MACID from the user-specified value to a system-generated value.

**Note:** The MAC address is used in LACP (Link Aggregation Control Protocol). It is necessary for proper operation of a virtual switch defined with the GROUP operand. The MAC address is also used when a z/VM TCP/IP stack is providing management services for a virtual switch, such as acting as an SNMP agent reporting bridge MIB information.

### **VEPA**

determines the operational mode of the virtual switch with regard to forwarding guest-to-guest and guest-to-external destination communications.

#### **OFF**

allows guest ports to communicate with each other and with any hosts and/or LPARs that share the same OSA port. VEPA OFF is the default setting for a QDIO virtual switch.

#### **ON**

prohibits guests from sending traffic to other guests on the same virtual switch without going through an external entity by forwarding all traffic from the guest through the OSA uplink to an adjacent switch. In addition, no direct LPAR communications sharing the same OSA port is permitted with the guest ports of the virtual switch. All traffic from the virtual switch destined for any sharing hosts/LPARs on the same OSA port will be forwarded as well. Any traffic destined for the virtual switch guest ports from hosts/LPARs sharing the same OSA port will also be forwarded to the adjacent switch. VEPA ON requires an ETHERNET virtual switch (without a Bridge Port) with OSA uplink(s) that supports VEPA as well as the partner switch must support Reflective Relay.

VEPA ON may not be specified if ISOLATION is ON.

VEPA ON may not be specified for a GLOBAL virtual switch.

### **ISOLation**

determines if guests on the virtual switch can communicate between themselves and other hosts and/or LPARs that share the same OSA-Express port with the virtual switch. ISOLATION OFF is set by default for all other virtual switch types to allow unrestricted network communications. ISOLATION is not supported for a virtual switch that has a Bridge Port.

#### **OFF**

allows guest ports to communicate with each other and with any hosts and/or LPARs that share the same OSA port.

#### **ON**

prohibits guests from sending traffic to other guests on the same virtual switch by discarding traffic that is destined for another guest port on the virtual switch. In addition, no direct LPAR communications sharing the same OSA port is permitted with the guest ports of the virtual switch. All traffic from the virtual switch destined for any sharing hosts/LPARs on the same OSA port will be dropped. Any traffic destined for the virtual switch guest ports from hosts/LPARs sharing the same OSA port will also be dropped.

ISOLATION ON may not be specified if VEPA is ON.

ISOLATION ON may not be specified for a GLOBAL virtual switch.

### **IPTimeout** *nnn*

indicates the length of time in minutes that a remote IP address table entry remains in the IP address table for the virtual switch.

*nnn* is a number from 1 to 240.

### **VLAN\_counters**

indicates the level of detail that should be maintained for the virtual switch network activity.

#### **OFF**

indicates that the detailed VLAN counters are not needed. That is, the system should stop keeping counters at the VLAN level, and begin keeping transmission counters only at the port level. Packet counters are kept for each port to show the number of frames received, transmitted or discarded. VLAN\_counters OFF is the default.

#### **ON**

indicates that the system should begin keeping transmission counters at the VLAN level. That is, for each port, packet counters are maintained to show the number of frames received, transmitted or discarded for each VLAN. See [Virtual Switch Configuration Strategy with VLAN in z/VM: Connectivity](#) for guidance on selecting VLAN IDs to maximize performance.

### **COUNTERS CLEAR [UPLINK|BRIDGEport] [ALL|DATA|DISCARDed|ERRORs]**

resets counts for the virtual switch.

#### **UPLINK**

resets counts for the uplink port.

#### **BRIDGEport**

resets counts for the bridge port.

#### **ALL**

resets discarded, error and data counts for the specified port.

#### **DATA**

resets data counts for the specified port. Data counts include transmitted packets, transmitted bytes, received packets and received bytes.

#### **DISCARDed**

resets discarded counts for the specified port.

#### **ERRORs**

resets error counts for the specified port.

### **QUEuestorage *numberM***

indicates the upper limit of the amount of fixed storage CP and Queued Direct I/O Hardware Facility will use for buffers for each OSA-Express data device.

*number* defines the maximum number of megabytes of storage that can be consumed for QDIO queues on a single OSA port. When multiple OSA devices are defined in a link aggregation group, then each OSA port within the group will use the specified amount of storage. Fixed storage is allocated as needed based on network traffic, until the maximum of *numberM* are allocated.

*number* is a number from 1 to 8. 8M is the default value.

### **GVRP**

specifies that the VLAN IDs in use on the virtual switch should be registered with GVRP-aware switches on the LAN. This allows offloading of VLAN filtering to the switch from the z/VM Control Program (CP).

### **NOGVRP**

Do not register VLAN IDs with GVRP-aware switches on the LAN.

### **NONrouter**

indicates that the OSA-Express device identified by the RDEV keyword will not act as a router to the virtual switch. If a datagram is received at this device for an unknown IP address, the datagram will be discarded. This is the default.

NONrouter is valid only for an IP virtual switch.

### **PRIrouter**

indicates that the OA Express device identified by the RDEV keyword will act as a primary router to the virtual switch. If a datagram is received at this device for an unknown IP address, the datagram will be passed to the virtual switch.

PRIrouter is valid only for an IP virtual switch.

### **PORTname portname**

is a 1- to 8-character name that identifies the OSA-Express adapter. You can specify a maximum of three port names. Multiple port names are used when different port names are needed for the multiple *rdevs* specified on the RDEV operand. See usage note “5” on page 1774 for more information.

**Note:** The PORTNAME operand is ignored when the GROUP operand is used.

### **IVLPORT**

specifies operations that are allowed only for an IVL-type virtual switch. An IVL virtual switch is a special type of virtual switch that provides communication infrastructure to exchange control information and data necessary to manage global networking objects that can span multiple systems running z/VM.

### **VLAN IVL\_vlanid**

provides the system administrator with a way to change the VLAN ID associated with the IVL port on an IVL virtual switch that is VLAN AWARE. This changes the default VLAN ID assigned when the IVL virtual switch was created. It is a number between 1 and 4094.

### **PING [MAC | SYSTEM | ALL]**

tests connectivity with another system running z/VM, to determine if it is a member of the same IVL domain.

#### **SYSTEM IVLsystemnm**

*IVLsystemnm* is the 8 byte EBCDIC z/VM system identifier of the remote system running z/VM as specified in the system configuration file.

#### **MAC IVLsystemMAC**

*IVLsystemMAC* can be a unicast or multicast MAC address. A MAC address can be entered using any one of the following MAC notations:

- Twelve contiguous hex character: 02A000123456
- Greater than twelve but no more than eighteen hex characters delineated by any non-space character. This permits both standard and even nonstandard MAC address notations to be entered as shown.
  - 02-A0-00-12-34-56
  - 02:A0:00:12:34:56
  - 02A000-123456

#### **ALL**

use ALL to send a ping to the IVL domain multicast MAC address which will check connectivity with all members of the IVL domain.

Use QUERY VMLAN to display the members of the IVL domain along with the *IVLsystemMAC* for each.

### **HEARTbeat TIMEout seconds**

changes the length of time that is allowed to expire before the local system declares a communication problem with another system in the IVL domain if heartbeat signals are missing. Every 5 seconds, each system in the IVL domain broadcasts a signal to confirm current communication with all IVL domain members.

The default HEARTBEAT TIMEOUT is 30 seconds.

The allowable range is 10 to 3600 seconds.

**RESET**

is intended to be used for diagnostic purposes when directed by IBM. The RESET operand:

- terminates and then recreates the IVL virtual switch's connection, forcing the system to serialize itself with all other members of the IVL domain.
- disconnects and immediately reconnects the IVL virtual switch UPLINK port.

**Usage Notes**

1. A virtual switch's connection to a real hardware LAN segment is not operational until an eligible z/VM TCP/IP stack is selected to be the controller for the OSA-Express device. CP selects an eligible z/VM TCP/IP stack to be the controller by either:

- It is recommended that the VSWITCH controller be at the same release level as CP, although all supported releases are allowed.
- If CONTROLLER *userid1* is specified on the DEFINE or SET VSWITCH commands or the DEFINE or MODIFY VSWITCH system configuration statements, with either a single user ID or a list of user IDs, only those user IDs are selected.
- If CONTROLLER \* is specified or allowed to default, CP selects from any eligible z/VM TCP/IP stacks.

A z/VM TCP/IP stack becomes eligible when:

- The TCP/IP MODULE running in the controller is at a release level that supports the function required for the virtual switch.
- An IUCV \*VSWITCH statement is included in its CP directory entry.
- The TCP/IP VSWITCH CONTROLLER statement is coded, and has defaulted to be ON or is explicitly set to ON in the TCP/IP configuration file or through an OBEYFILE command.
- The stack has completed initialization.
- The stack has virtual device numbers available for CP to attach the control device.

The virtual device range used by CP is specified in the VSWITCH CONTROLLER TCP/IP configuration statement. If no VDEV range is specified, CP uses the virtual device number (*vdev*) that matches the *rdev* number specified on the DEFINE VSWITCH or SET VSWITCH command. See *z/VM: TCP/IP Planning and Customization* for more information about the [VSWITCH CONTROLLER Statement](#).

**Note:** Do not code DEVICE and LINK TCP/IP configuration statements for the device. Do not attach the device to a TCP/IP controller virtual machine. These steps are handled by DEFINE VSWITCH processing when a controller is selected.

If an eligible stack is not found, or none of the *rdevs* are operational, you receive a message, and the virtual switch operates in a local LAN environment.

2. Use the QUERY CONTROLLER command output to find the z/VM TCP/IP stacks that are the virtual switch controllers. Use the QUERY VSWITCH command to display information about virtual switch.
3. CP manages the devices used to control a virtual switch's connection to a real LAN segment through an OSA-Express device. CP attaches the devices to the z/VM TCP/IP virtual machine. CP also defines a device of type VSWITCH-OSD to the z/VM TCP/IP stack, concatenating *switchname* with *vdev* and "DEV" to form the device name and *switchname* with *vdev* and "LINK" to form the link names. These names appear in the TCP/IP query and trace information. The status of a link that represents the active connection to a real LAN segment is **Status: Ready**. **Status: Inactive** is shown for any links defined for backup devices.

DEVICE and LINK statements must not be included in the TCP/IP configuration file for these devices. Use the CONNECT and DISCONNECT operands to control whether the traffic flows through the OSA-Express Adapter to a real LAN segment.

If the OSA-Express Adapter has been initialized, you must issue SET VSWITCH with the DISCONNECT option before the PORTNAME, CONTROLLER, NONROUTER, PRIROUTER, GVRP, VEPA, PRIQUEUING,

or the currently active UPLINK port values can be changed. However, you can add or remove backup RDEVs while the virtual switch is connected.

4. Multiple real devices and portnames can be specified on the RDEV and PORTNAME parameters. If your OSA-Express device requires a portname, specify one portname for each real device number. This feature allows failover to an alternate real device in the event of a failure with the current OSA-Express device or link aggregation group. All real devices specified must be active and connected in order to dynamically failover to an alternate device. In addition, the alternate devices must be defined on separate CHPIDs.
5. When the real device identified by one of the *rdevs* is started on a VSWITCH command with the NOGROUP attribute, TCP/IP assigns the port name as the hardware adapter name. If an adapter name was already assigned by a previous connection, then the port name must be the same as assigned by any other connection in order to share the adapter. This includes sharing the OSA-Express adapter with this logical partition or all other partitions.

The PORTNAME operand is optional. However, some levels of the OSA-Express adapters require that the PORTNAME operand is specified. When such an adapter is in use and the PORTNAME operand is omitted, an error message is displayed during switch initialization.

If the device is already started, you must stop it by issuing the SET VSWITCH *switchname* DISCONNECT command before changing the port name.

6. VLAN configuration information can be found in [Planning for Guest LANs and Virtual Switches in z/VM: Connectivity](#).
7. PRIROUTER is required only when IP forwarding (routing) nodes will be coupled to the switch. Router nodes provide connectivity for their LAN segments (remote nodes) through their switch connection. When Router nodes are deployed, their switch connection must be configured as PRIROUTER. In addition to this, the switch itself must also be configured as PRIROUTER to the OSA-E adapter. This will insure delivery of datagrams destined for LAN segments that are connected through routers coupled to a switch. Only one connection on each OSA-Express card can be designated as PRIROUTER. If the switch is successful in establishing PRIROUTER on the OSA-Express card, no other node (or switch) sharing the same OSA-Express<sup>®</sup> card will be able to act as PRIROUTER. If another connection has already been established as PRIROUTER, the switch will be left with NONROUTER status (which is reflected in the QUERY VSWITCH response).
8. NONROUTER is the default mode for the switch. Every node is directly coupled to the switch and the associated IP destinations are registered with the OSA-Express connection. This is the most efficient way to use the virtual switch. In this mode, packets with an unrecognized IP destination are automatically sent out through the switch connection.
9. If the virtual switch has a guest port for the UPLINK port (UPLINK NIC *userid2 vdev*), the PRIROUTER and NONROUTER options are ignored.
10. If the virtual switch has been defined as VLAN UNAWARE, the following are not allowed:
  - the PORTTYPE and VLAN option on GRANT
  - the PORTTYPE and VLAN option on PORTNUMBER
  - the VLANID option
11. If a virtual switch is defined as VLAN aware and no VLAN membership is given with the GRANT, PORTNUMBER, NICDEF or the VLANID (or provided by an external security manager), a guest is a member of the virtual switch's default VLAN ID (*defvid*) only if a *defvid* was assigned as part of the virtual switch definition.
12. The MACID may not be changed while a z/VM TCP/IP stack is providing management (SNMP) services for the virtual switch. That is, when a HOME TCP/IP configuration statement has established a management IP address for the virtual switch using the VSWITCH *switchname* keyword. Use the QUERY VSWITCH command to find the name of the stack providing management services and contact the administrator of the stack to remove the management IP address using the OBEYFILE command for the HOME configuration statement. This disables management services (including SNMP) for the virtual switch. When the MACID has been changed successfully notify SNMP clients

of the MACID change and issue the OBEYFILE command to reassign the IP address and reactivate SNMP services.

13. The MACID may not be changed while the virtual switch connection to the hardware LAN is active. Use the SET VSWITCH *switchname* DISCONNECT command before changing the MACID.
14. Changing the RDEV list with SET VSWITCH RDEV can affect SNMP data, specifically the port number that represents the OSA-E connection to the real network. This is the Bridge MIB *dot1dBasePort* variable (1.3.6.1.2.1.17.1.4.1).
15. Using the VLAN\_counters ON option on SET VSWITCH can significantly degrade throughput and increase system utilization for a virtual switch. Use this option only if there is a need to understand traffic patterns at the level of VLAN IDs. One example of when you might want VLAN\_counters ON is when you are using a network management tool to analyze VLAN network traffic.
16. Detailed transmission counters are available only for a VLAN AWARE Virtual Switch with the VLAN\_counters option set to ON. These values can be obtained by using the Diagnose x'26C' interface.
17. If the virtual switch is VLAN aware, the VLANs need to be configured with the GRANT, VLANID or PORTNUMBER command or via the NICDEF statement in the user directory. If an External Security Manager (ESM) is in control of the virtual switch, the VLANs configured (with GRANT, VLANID, PORTNUMBER or NICDEF) must be included in the VLAN list provided by the ESM.
18. When SET VSWITCH commands are used in conjunction with NICDEF statements to configure the network attributes, the following rules apply:
  - No SET VSWITCH configuration is required if the NICDEF statement provides all necessary network configuration.
  - NICDEF attributes override any prior SET VSWITCH configuration (and this is reflected in subsequent QUERY VSWITCH output).
  - After a device is connected to the virtual switch, subsequent SET VSWITCH commands change the active configuration (but do not alter NICDEF statements in the USER DIRECT source).
  - Each time the virtual NIC is created (or coupled) to the network, network attributes from the NICDEF statement are refreshed (replacing any dynamic changes made using the SET VSWITCH command).
19. The following notes are related to the SET VSWITCH PATHMTUDISCOVERY option:
  - When a PORT GROUP is defined using the SET PORT command, the PATHMTUDISCOVERY option applies to all members of the group.
  - If a virtual switch is configured with an UPLINK NIC, the SET VSWITCH PATHMTUDISCOVERY option is not applicable. However, it can be set for use later when an UPLINK RDEV port device is defined.
  - If a virtual switch is configured with the NOUPLINK option, the SET VSWITCH PATHMTUDISCOVERY option is not allowed.
20. A Global virtual switch is a collection of virtual switches that share the same networking characteristics. This collection of virtual switches spans multiple systems running z/VM but logically operates as a single switch.

Virtual switches defined with the same name and the GLOBAL option are said to be members of the same global virtual switch when they reside in systems running z/VM that belong to the same IVL domain. (Definition and activation of an IVL virtual switch allows a system running z/VM to join an IVL domain. See usage note [“19” on page 243](#) for more information about the IVL virtual switch.)

The following conditions must be fulfilled in order to create or change a global virtual switch member:

- The system's IVL virtual switch must be defined and its UPLINK port connected. In other words, the system must be an active member of an IVL domain.

Creation of Global virtual switches specified in the SYSTEM CONFIG file will be deferred until the IVL virtual switch UPLINK port connects the system to the IVL domain. Message HCP3178I will be displayed for each deferred Global virtual switch.

## SET VSWITCH

- If any other virtual switch exists in the IVL domain with the same name the following attributes must match or the DEFINE VSWITCH or SET VSWITCH will fail with message HCP3170E.
  - IP or ETHERNET
  - ISOLATION
  - VEPA
  - VLAN AWARE or UNAWARE
  - NATIVE *natvid*
  - USERBASED or PORTBASED
- The following additional restrictions exist if a global virtual switch is configured to use a shared port group:
  - RDEV *device\_addr* is not allowed

21. SET VSWITCH COUNTERS CLEAR only affects the data displayed via the QUERY VSWITCH and QUERY NIC commands. It does not affect other data such as Virtual Switch monitor records, Diagnose x'26C' data, or SNMP BRIDGE-MIBs for a virtual switch.

### Examples

The following examples apply when CP, not an ESM, is protecting the guest LAN:

1. To allow user VMTCPIP to connect to virtual switch BIGANG and filter any incoming traffic using VLAN 9, specify the following:

```
set vswitch bigang grant vmtcpip vlan 9
```

2. To remove the authority of LINUX1 to connect to virtual switch BIGANG, specify the following:

```
set vswitch bigang revoke linux1
```

3. To connect virtual switch BIGANG to a real LAN using device FD00, specify the following:

```
set vswitch bigang rdev fd00
```

4. To disconnect virtual switch BIGANG from device FD00 and reconnect it to a real LAN using device FE00 and portname ableport, specify the following:

```
set vswitch bigang disconnect
set vswitch bigang portname ableport
set vswitch bigang rdev fe00
set vswitch bigang connect
```

5. To define a port based virtual switch VSWITCH1 with multiple access ports 10 and 20 for user1, specify the following. Port 10 will be added to VLAN 11 and port 20 will be added to VLAN 21.

```
define vswitch vswitch1 portbased vlan aware
set vswitch vswitch1 portnumber 10 userid user1
set vswitch vswitch1 portnumber 20 userid user1
set vswitch vswitch1 vlanid 11 add 10
set vswitch vswitch1 vlanid 21 add 20
```

6. Given VSWITCH1 (a VLAN AWARE network), individual device connections can be configured in the user directory with no need for SET VSWITCH commands:

```
USER USER1 ...
//
NICDEF 2100 TYPE QDIO LAN SYSTEM VSWITCH1 VLAN 11 PORTTYPE ACCESS
NICDEF 2200 TYPE QDIO LAN SYSTEM VSWITCH1 VLAN 21 PORTTYPE ACCESS
```

### Responses

Response 1:



```
command complete
```

is the normal response from this command.

## Messages

- HCP002E Invalid operand - *operand*.
- HCP003E Invalid option - *option*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP025E Storage missing or invalid
- HCP2768E Missing switchname
- HCP2783E VSWITCH SYSTEM *switchname* does not exist
- HCP2798I User *userid* is already authorized
- HCP2799E VSWITCH change is not allowed.
- HCP2838I Device *device* specified for VSWITCH *switchname* BRIDGEPORT is not a bridge-capable HiperSockets device.
- HCP2838I Device *device* specified for VSWITCH *switchname* does not exist.
- HCP2838I Device *device* specified for VSWITCH *switchname* is already attached to user *username*.
- HCP2838I Device *device* specified for VSWITCH *switchname* is not compatible with the VSWITCH type.
- HCP2838I Device *device* specified for VSWITCH *switchname* is offline.
- HCP2838I Port *portnumber* specified for device *device* is invalid; maximum port number is *maxportnumber*.
- HCP2838I Device *device* specified for VSWITCH *switchname* is not configured for priority queuing.
- HCP2839E Invalid operand for specified VSWITCH: *operand*
- HCP2846E The [VLAN | PORTTYPE | GVRP | NATIVE | VLAN\_COUNTERS | VLANID | IVLPORT VLAN] option is not allowed for a VLAN UNAWARE virtual switch.
- HCP2847E PORTTYPE ACCESS is not allowed when the user is authorized for more than one VLAN
- HCP2850E MACID is not set for *switchname*. SSI cluster mode not stable.
- HCP2850E MACID is not set for VSWITCH *vswname*. [ Stack *userid* is providing management services | MAC address *macaddr* is already in use [ by *user ID* ] | Suffix for MAC address *macaddr* is not within USER MACIDRANGE | No MACID is available ].
- HCP2853E SET VSWITCH not performed; device limit exceeded.
- HCP2856E SET VSWITCH not performed; port group already assigned to VSWITCH SYSTEM *switchname*.
- HCP2856E SET VSWITCH not performed; port group *groupname* is already in use.
- HCP2858I Guest *userid* is authorized for too many VLAN IDs on SYSTEM *switchname*. VLAN-specific ARP processing is disabled.
- HCP2858I *userid* connection to *lanowner lanname* has been [revoked | updated] by *userid1*.
- HCP2858I *userid* connection to *lanowner lanname* is not affected by this update.
- HCP3001E Device *device* is defined for both port group *groupname* and VSWITCH SYSTEM *switchname*.
- HCP3002E Unable to establish ISOLATION mode at this time; VSWITCH SYSTEM *switchname* configuration in transition.
- HCP3003E Controller *controllername* for VSWITCH SYSTEM *switchname* does not support OSA QDIO Connection Isolation.
- HCP3004E Device *device* for VSWITCH SYSTEM *switchname* does not support OSA QDIO Connection Isolation.

## SET VSWITCH

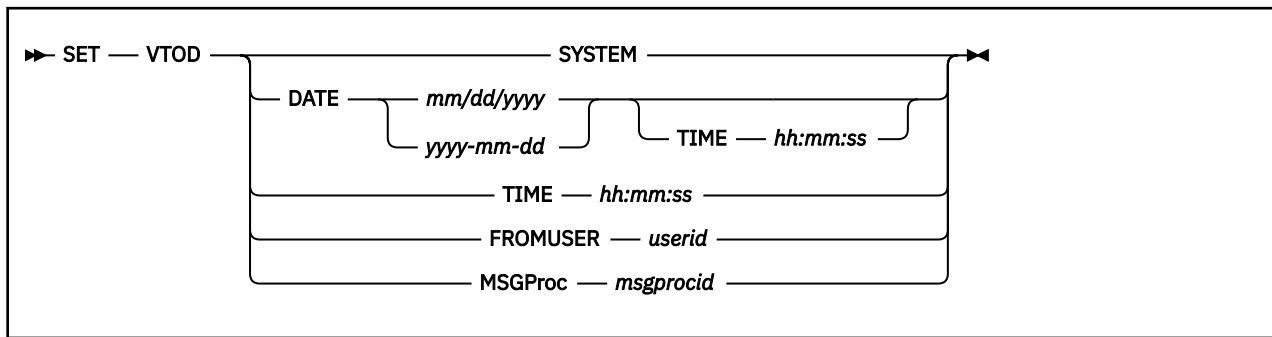
- HCP3022E SET VSWITCH not performed; virtual switch *switchname* is defined with NOUPLINK.
- HCP3022E SET VSWITCH not performed; user *userid* not logged on.
- HCP3022E SET VSWITCH not performed; device *vdev* does not exist.
- HCP3022E SET VSWITCH not performed; NIC type *nictype* does not match VSWITCH type *switchname*.
- HCP3022E SET VSWITCH not performed; NIC *vdev* is already attached to *owner switchname*.
- HCP3022E SET VSWITCH not performed; CONNECT or DISCONNECT is already in progress on the {UPLINK | BRIDGE} port.
- HCP3022E SET VSWITCH not performed; *userid* is not authorized to COUPLE to VSWITCH *switchname*.
- HCP3022E SET VSWITCH not performed; VLAN configuration error.
- HCP3022E SET VSWITCH not performed; NIC transport type *transport1* does not match VSWITCH transport type *transport2*.
- HCP3022E SET VSWITCH not performed; internal logic error code *code*.
- HCP3022E SET VSWITCH not performed; *vdev* is not the correct device type.
- HCP3022E SET VSWITCH not performed; VSWITCH does not exist.
- HCP3022E SET VSWITCH not performed; ESM error.
- HCP3022E SET VSWITCH not performed; Uplink port error.
- HCP3022E SET VSWITCH not performed; ESM VLAN configuration error.
- HCP3022E SET VSWITCH not performed; MACPROTECT is not valid for the IP transport type.
- HCP3022E SET VSWITCH not performed; [BRIDGEPORT | VEPA] is not valid when the transport type is IP.
- HCP3022E SET VSWITCH not performed; [BRIDGEPORT | VEPA] is not valid when the ISOLATION setting is ON.
- HCP3022E SET VSWITCH not performed; ISOLATION is not valid when a Bridge Port is defined.
- HCP3022E SET VSWITCH not performed; BRIDGEPORT RESET requires the virtual switch to be the active Bridge Port connection.
- HCP3022E SET VSWITCH not performed; PATHMTUDISCOVERY is not valid when the virtual switch is defined with the NOUPLINK option.
- HCP3022E SET VSWITCH not performed; [VEPA | ISOLATION] is not valid when [ISOLATION | VEPA] is on.
- HCP3022E SET VSWITCH not performed; SWITCHOVER is not valid when the virtual switch is disconnected.
- HCP3022E SET VSWITCH not performed; SWITCHOVER is not valid when the uplink port is a NIC.
- HCP3022E SET VSWITCH not performed; SWITCHOVER is allowed only when there is an eligible backup device in the uplink RDEV list.
- HCP3022E SET VSWITCH not performed; SWITCHOVER is allowed only when the active device is in the uplink RDEV list.
- HCP3022E SET VSWITCH not performed; SWITCHOVER is allowed only when the virtual switch is not configured for link aggregation.
- HCP3022E SET VSWITCH not performed; UPLINK NIC is not valid when the VEPA setting is ON.
- HCP3022E SET VSWITCH not performed; VEPA is not valid when an Uplink NIC is defined.
- HCP3022E SET VSWITCH not performed; [VEPA | ISOLATION] ON is not valid for a GLOBAL virtual switch.
- HCP3022E SET VSWITCH not performed; UPLINK NIC is not valid for an IVL virtual switch.
- HCP3022E SET VSWITCH not performed; No port is available.
- HCP3022E SET VSWITCH not performed; PRIQUEUING OFF is not valid for an IVL virtual switch.
- HCP3022E SET VSWITCH not performed; PQUPLINKTX is not valid for an IVL virtual switch.

- HCP3031I Port *portnum* cannot join VLAN *vlanid*. The port is not defined.
- HCP3031I Port *portnum* cannot join VLAN *vlanid*. Access ports cannot join more than one VLAN.
- HCP3032I Port *portnum* cannot be removed from VLAN *vlanid*. The port is not defined.
- HCP3032I Port *portnum* cannot be removed from VLAN *vlanid*. The port has not joined that VLAN.
- HCP3033E SET VSWITCH not performed; Port *portnum* is already defined.
- HCP3033E SET VSWITCH not performed; Port *portnum* is not defined.
- HCP3033E SET VSWITCH not performed; VLAN *vlanid* is not defined.
- HCP3033E SET VSWITCH not performed; Port definition conflicts with Uplink NIC port definition.
- HCP3033E SET VSWITCH not performed; Uplink NIC port conflicts with existing port definition.
- HCP3039I Port *portnum* has joined VLAN *vlanid*.
- HCP3039I Port *portnum* has been removed from VLAN *vlanid*.
- HCP3040I SET VSWITCH completed with messages.
- HCP3045E The Bridge facility is not supported by this processor.
- HCP3046E Port number conflicts with configured port *portnum*.
- HCP3049E An internal error occurred; the state of the virtual switch *switchname* may be unstable.
- HCP3160E Virtual switch *switchnm* cannot be configured with the RDEV operand specifying back up devices.
- HCP3160E Virtual switch *switchnm* cannot be configured with a shared port group.
- HCP3161E [DEFINE VSWITCH | SET VSWITCH] is not performed; a port group defined with the SHARED option can be used only with a virtual switch defined with the GLOBAL option.
- HCP3163E SET VSWITCH is not performed; IVLPORT is valid only for an IVL virtual switch.
- HCP3163E SET VSWITCH is not performed; system *sysname* is isolated from IVL Domain *domain* due to a configuration problem.
- HCP3163E SET VSWITCH is not performed; the IVL virtual switch does not exist or its IVL port is not connected.
- HCP3166E [DETACH VSWITCH | SET VSWITCH DISCONNECT] for IVL VSWITCH *ivlswitchnm* failed. Virtual switch *switchname* is defined on *systemnm* with the GLOBAL attribute.
- HCP3167E [DEFINE VSWITCH *switchnm* | SET VSWITCH *switchnm* | SET PORT GROUP *groupname*] failed. The IVL virtual switch does not exist or its UPLINK port is not connected.
- HCP3170E VSWITCH *systemname.switchname* [is not created | is not changed]. The virtual switch defined on *systemname* [ conflicts with the [ IP | ETHERNET | ISOLATION | VEPA | VLAN AWARE | VLAN UNAWARE | NATIVE VLAN | USERBASED | PORTBASED ] attribute ].
- HCP3170E VSWITCH *systemname.switchname* [is not created | is not changed]. An internal error occurred.
- HCP3170E VSWITCH *systemname.switchname* [is not created | is not changed]. The virtual switch defined on *systemname* [ is currently not responding | is currently in synchronization ].
- HCP3173E Device *device* is reserved for system use.
- HCP3174I System *systemname* is not a member of the local system's IVL Domain.
- HCP3186E [DEFINE VSWITCH GROUP | SET PORT GROUP | SET VSWITCH GROUP] for shared port group *groupname* not performed; devices on OSA PCHID *pchid* do not exist or are not available.
- HCP3195E SET [VMLAN | VSWITCH] not performed; trace table can not be allocated.
- HCP3196E SET VSWITCH not performed; the port group must be defined before associating it with a GLOBAL vswitch.
- HCP3199E SET VSWITCH not performed; uplink port group *groupname* on this system is not equivalent to other IVL members sharing the same port group.
- HCP3221E SET VSWITCH GROUP not performed; usage of shared port group *groupname* would exceed the maximum number of virtual switches allowed by a single system sharing the same port group.

## SET VSWITCH

- HCP6704E Missing token at end of line.
- HCP6706E Invalid BUFFER value - *value*
- HCP6706E Invalid device address - *vdev*
- HCP6706E Invalid device address or port number - *{vdev | portnumber}*
- HCP6706E Invalid groupname - *groupname*
- HCP6706E Invalid HEARTBEAT value - *value*
- HCP6706E Invalid MACID - *id*
- HCP6706E Invalid number - *number*
- HCP6706E Invalid PATHMTUDISCOVERY value - *value*
- HCP6706E Invalid PING value - *value*
- HCP6706E Invalid portname - *portname*
- HCP6706E Invalid port number - *portnumber*
- HCP6706E Invalid switchname - *switchname*
- HCP6706E Invalid TRACESIZE value - *value*
- HCP6706E Invalid VLAN id - *vlanid*

## SET VTOD



### Authorization

Privilege Class: A, B, G

### Purpose

Use the SET VTOD command to set a virtual machine's Time-of-Day (TOD) clock. By default, the virtual machine's TOD clock is set and synchronized to the system TOD clock at LOGON. This command:

- enables a user to set their virtual TOD clock to a different date and time,
- enables a user to synchronize the virtual TOD clock of a message processor user to the issuing user's virtual TOD clock,
- enables a user to synchronize their virtual TOD clock with the virtual TOD clock of another user ID,
- resets the user's virtual TOD clock to the system TOD clock.

Use of this command requires authorization which is provided by the OPTION directory statement. Refer to the individual operand description below and the [OPTION Directory Statement](#) in *z/VM: CP Planning and Administration*.

### Operands

#### SYSTEM

the user's virtual machine TOD clock will be set to the current system value. This is the default value that is set at LOGON.

#### DATE *mm/dd/yyyy*

#### DATE *yyyy-mm-dd*

specifies the new date to be used for this virtual machine's TOD clock. If the TIME operand is not specified, the date is changed and the time remains unchanged. The date entered must be in a range from 01/01/1942 to 12/31/2041.

#### TIME *hh:mm:ss*

specifies the new time to be used for this virtual machine's TOD clock. If the DATE operand is not specified, the time is changed and the date remains unchanged.

#### FROMUSER *userid*

the user's virtual machine TOD clock will be set exactly to the specified user ID's TOD clock.

The user must be authorized to use this operand via the TODENABLE operand of the OPTION directory statement. Without authorization, the virtual machine's TOD clock will not be changed and an error message is presented.

#### MSGPROC *msgprocid*

the virtual TOD clock of the specified message processor user will be set exactly to the issuer's virtual TOD clock.

## SET VTOD

If you have Class G privilege only, the message processor that you specify must be currently in your I/O configuration. A message processor may be defined in your I/O configuration by either the DEFINE MSGPROC command or the SPECIAL directory statement.

If you have Class A or Class B privilege, the specified message processor user must have the CFVM operand of the OPTION directory statement.

Regardless of privilege class, the specified message processor user must have the TODENABLE operand of the OPTION directory statement.

**CAUTION:** Setting the TOD clock of the message processor will cause a RESTART MSGPROC command to be executed on the specified message processor. The RESTART MSGPROC command will cause the message processor to be re-IPLed, thus causing all structures within the Coupling Facility to be lost and will affect all coupled guests connected to the message processor. See the RESTART MSGPROC command for more information.

### Usage Notes

1. A system reset will occur for the virtual machine whose virtual TOD clock value has changed. If the virtual TOD clock of a message processor is changed, the message processor's virtual machine is restarted.
2. The virtual TOD clock will be changed for all virtual CPUs of the affected configuration.
3. This command does not affect the time and date information that is returned from DIAG X'000C', DIAG X'270', REXX DATE(), REXX TIME(), or PIPE TIMESTAMP functions. The EXEC2 &TIME and &DATE functions are affected. This command is NOT intended to allow an ordinary CMS user to run their virtual machine with a date and time that is different than the system date and time. This command is to allow guest operating systems running on VM to synchronize their virtual TOD clocks in order to participate in a parallel sysplex environment.

### Responses

Response 1:

To modify the date of your virtual machine's TOD clock, enter the following:

```
set vtod date 12/31/1999
Storage cleared - system reset.
```

Response 2:

To set the TOD clock of the message processor CFCC2 to your virtual machine's TOD clock, enter the following:

```
set vtod msgp cfcc2
HCPMFM2809I Restart of message processor CFCC2 by SHARE1 initiated.
```

Message processor CFCC2 will automatically be restarted. HCP2809I will be displayed to both the message processor and the user who issued the SET VTOD MSGP CFCC2 command.

### Messages

- HCP002E Invalid operand – *operand*
- HCP003E Invalid option – command contains extra option(s) starting with *option*
- HCP020E Userid missing or invalid
- HCP045E *userid* not logged on
- HCP1014E A required option is missing.
- HCP2805E Message processor *msgprocid* does not exist in your I/O configuration.
- HCP2809I Restart of message processor *msgprocid* by *userid* initiated.

- HCP2812E User *userid* does not have *operand* authorization on *statement* directory statement
- HCP6704E Missing token at end of line
- HCP6706E Invalid string – *string*
- HCP6706E Invalid time – *time*
- HCP6706E Invalid date – *date*

## SET WNG

---



### Authorization

Privilege Class: G

### Purpose

Use SET WNG to control the display of WARNING command messages. The initial setting for WARNING is installation dependent and is defined in the system configuration file.

### Operands

#### ON

#### OFF

determines whether warning messages are shown on your display screen. If you specify ON, all messages sent using the WARNING command are displayed. If you specify OFF, no warning messages are received.

#### IUCV

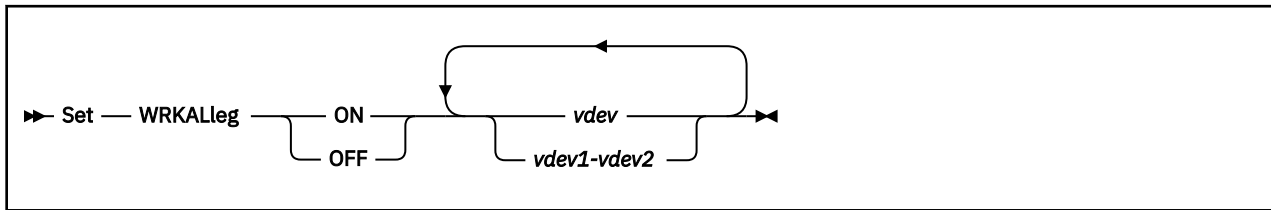
specifies that warnings are to be passed to the virtual machine through IUCV if a connection to the message system service exists. If no IUCV connection exists, the message is handled as if SET WNG ON had been issued.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid



## SET WRKALLEG



### Authorization

Privilege Class: G

### Purpose

Use SET WRKALLEG to turn virtual working allegiance simulation on or off for one or more minidisks. Working allegiance guarantees that if multiple guests (or one guest using multiple virtual devices) start I/O to the same device, one channel program will run to completion or be terminated before another channel program is allowed to begin. When WRKALLEG is off, CP may interleave execution of guest channel programs to the same minidisk.

**Note:** You must have a write link to the specified minidisks to issue this command.

### Operands

#### ON

indicates that virtual working allegiance will be activated for the minidisks specified.

#### OFF

indicates that virtual working allegiance will be ended for the minidisks specified.

#### vdev

#### vdev1-vdev2

is the virtual device number or a range of virtual device numbers of the minidisks.

### Usage Notes

1. This command must be used when running two or more z/OS guests as part of a Sysplex configuration using the cross-system coupling facility (XCF) component of z/OS. SET WRKALLEG ON must be used for any minidisk containing the XCF couple dataset to maintain cross-system lock integrity (and thereby, data integrity) within the sysplex.
2. VM supports a z/OS guest's participation in a Sysplex configuration only if all participating z/OS systems are guests of the same VM host. Environments which include a z/OS system running natively in a separate processor partition, or as a guest of a different VM system are not supported.
3. This command is rejected for full-pack minidisks on a shared real volume. Virtual working allegiance is incompatible with sharing real DASD. SET WRKALLEG is only for sharing virtual devices.
4. This command is rejected for dedicated DASD. Working allegiance is only necessary for shared minidisks. Virtual working allegiance is incompatible with sharing real DASD. SET WRKALLEG is only for sharing virtual devices.
5. Working allegiance applies only to users with write access to the minidisk. This command is rejected if the issuer has read-only access. Furthermore, working allegiance is simulated only when a guest with write access initiates I/O. Only one guest of the sysplex configuration needs to issue SET WRKALLEG ON to activate the function.
6. Working allegiance is always simulated for virtual disks in storage. SET WRKALLEG is rejected if the vdev is a virtual disk in storage.

## SET WRKALLEG

7. The WRKALLEG setting applies to the minidisk, not to an individual virtual device. If two users have links to the same minidisk and one issues SET WRKALLEG, the setting will apply to both users' access to the minidisk.
8. If WRKALLEG is turned on and I/O is already in progress at the minidisk, the first working allegiance I/O to the minidisk may be interleaved with the previous I/O.
9. Caution must be used when WRKALLEG is turned off. If WRKALLEG is turned off while an I/O is already in progress, unpredictable results may occur. The issuer should determine that all users to the minidisk no longer need working allegiance before terminating the function.
10. For more information on working allegiance, see the Principles of Operation for your processor.
11. For information about the WRKALLEG rules when linking, see [“LINK” on page 578](#).

## Responses

Response 1:

```
DASD <vdev> working allegiance is being
simulated among same system guests.
```

is returned for each virtual device specified where WRKALLEG=ON.

Response 2:

```
DASD <vdev> working allegiance is turned off.
```

is returned for each virtual device specified where WRKALLEG=OFF.

## Messages

- HCP003E Invalid option - *option*
- HCP006E Invalid device type - *vdev*
- HCP026E Operand missing or invalid
- HCP040E Device *vdev* does not exist
- HCP2010E DASD *vdev* is dedicated, WRKALLEG is rejected.
- HCP2010E DASD *vdev* is read only, WRKALLEG is rejected.
- HCP2010E DASD *vdev* is a full-pack mini on a shared real volume, WRKALLEG is rejected.
- HCP2010E DASD *vdev* is a VDISK, WRKALLEG is rejected.
- HCP6645E SSI cluster is in safe mode; command rejected.
- HCP6706E Invalid address range - *string*

## SET 370ACCOM



### Authorization

Privilege Class: G

### Purpose

Use SET 370ACCOM to allow System/370 CMS applications to run in an ESA/390 or ESA/XC virtual machine.

### Operands

#### ON

indicates that 370-only applications can run in an ESA/390 or ESA/XC virtual machine.

#### OFF

indicates that 370-only applications cannot run in an ESA/390 or ESA/XC virtual machine. This is the initial setting.

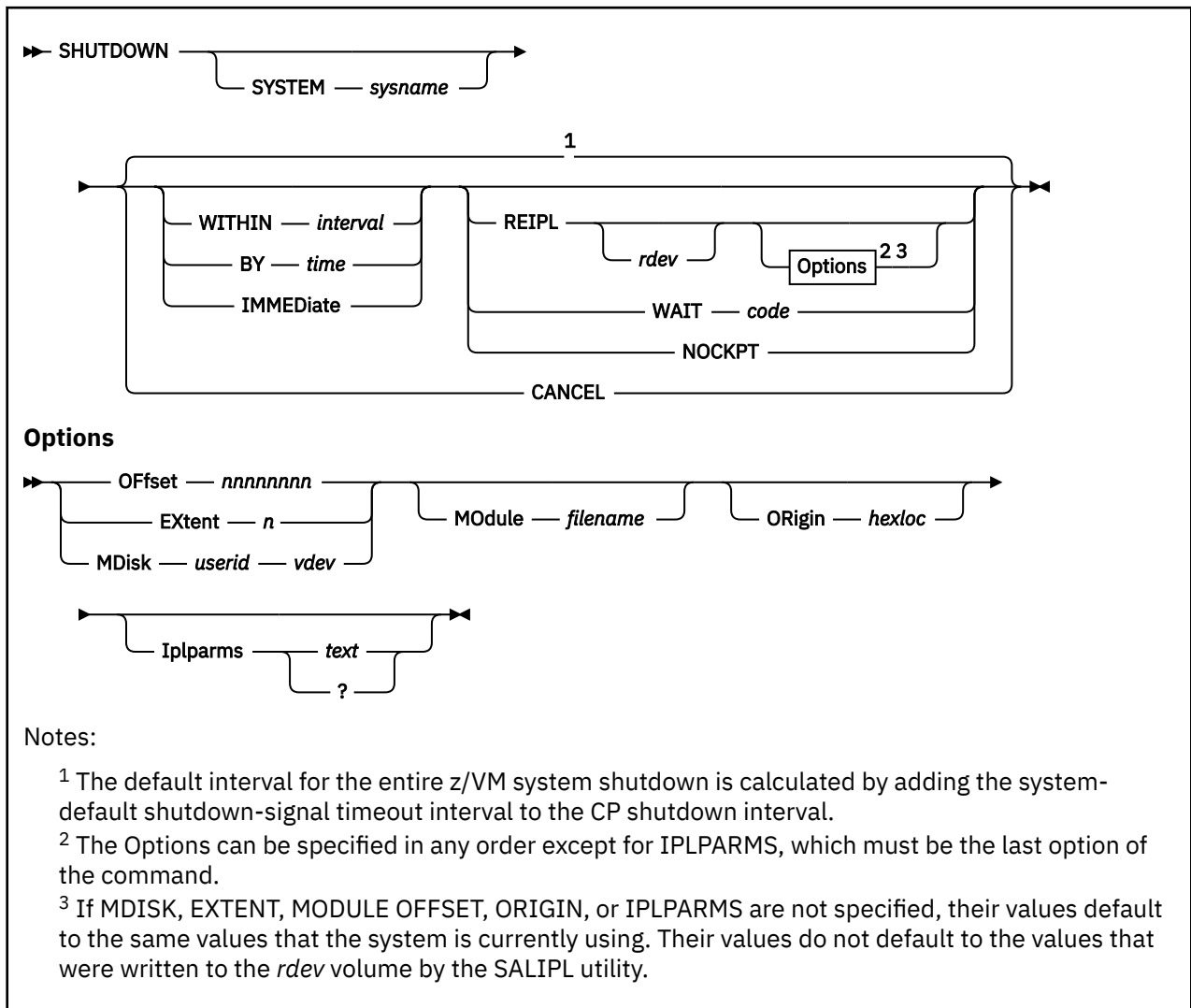
### Usage Notes

1. When 370ACCOM is set ON, it remains on until it is either set OFF or until the virtual machine experiences a subsystem-reset operation. A subsystem-reset operation occurs during the execution of commands such as the following:
  - IPL
  - DEFINE STORAGE
  - SET MACHINE
  - SYSTEM RESET
  - SYSTEM CLEAR
  - DETACH CPU
2. A guest in z/Architecture or z/XC mode can not have 370ACCOM ON.
  - If a MACHINE ESA guest with 370ACCOM ON switches into z/Architecture mode, 370ACCOM will be turned OFF as part of the transition.
  - If a MACHINE XC guest with 370ACCOM on switches to z/XC mode, 370ACCOM will be turned off as part of the transition.
  - If a MACHINE Z guest issues the SET 370ACCOM ON command, the command will be rejected with error message HCP2632E.
3. For information on running your 370-only CMS applications in an ESA/390 or ESA/XC virtual machine, see the following topics in *z/VM: CP Programming Services*:
  - [370 Accommodation Facility Overview](#)
  - [370 Accommodation Facility Definition](#)
  - [DIAGNOSE Code X'58' – 3270 Virtual Console Interface](#)

**Messages**

- HCP2632E When a virtual machine is in z/Architecture or z/XC mode, you cannot set 370ACCOM ON.

# SHUTDOWN



## Authorization

Privilege Class: A

## Purpose

Use SHUTDOWN to complete the following tasks:

- Systematically end all system function.
- Checkpoint the system for an eventual warm start.

Optionally, SHUTDOWN does an automatic warm start of the current or specified CP module.

## Operands

### SYSTEM *sysname*

specifies the name of the system that is to be shut down. The *sysname* must be the name of the system on which the command is issued.

## SHUTDOWN

The SHUTDOWN command shuts down only the system on which the command is issued. Specifying the system name serves as a check to verify that the intended system is being shut down (you cannot shut down system B from system A).

The system configuration file can be defined to require the SYSTEM to be specified on the SHUTDOWN command (see the [FEATURES Statement](#) in *z/VM: CP Planning and Administration*). This might be preferred in large data centers with multiple operator consoles to help prevent accidentally shutting down the wrong system.

### **WITHIN interval**

specifies a time interval for the entire z/VM system shutdown.

The *interval* specifies a number of seconds in the range 1 - 65535.

If the interval is greater than the CP shutdown interval, CP immediately sends a shutdown signal to enabled guests and delays the start of the CP shutdown. The start of the CP shutdown is delayed until either of the following events occurs:

- All signaled guests indicate that they shut down.
- The shutdown-signal timeout interval elapses.

See usage notes [“10” on page 1792](#), [“11” on page 1792](#), and [“12” on page 1793](#).

### **BY time**

specifies a time-of-day target for the entire z/VM system shutdown.

The *time* can be specified as *hh:mm* or *hh:mm:ss*. The interval from the time of day that the command is entered until the specified time-of-day must be in the range 1 - 65535 seconds.

The start of the CP shutdown is delayed until one of the following events occurs:

- All signaled guests indicate that they shut down.
- The shutdown-signal timeout interval elapses.
- The specified time of day (minus the amount of time that is reserved for a CP shutdown) is reached.

See usage notes [“10” on page 1792](#), [“11” on page 1792](#), and [“12” on page 1793](#).

### **IMMEDIATE**

shuts down the entire z/VM system immediately without sending shutdown signals to enabled users, even if a previous SHUTDOWN command is pending. If a previous SHUTDOWN command is pending, its operands are not used and must be specified with IMMEDIATE on the new command if they are required. See usage note [“10” on page 1792](#).

### **REIPL**

specifies that the system is to be restarted immediately after the SHUTDOWN command completes.

#### **rdev**

identifies the real device to be IPLed when the system is restarted. If *rdev* is not specified, the system will REIPL from the *rdev* that was used for the last IPL or SHUTDOWN REIPL.

**Note:** If you specify *rdev*, you cannot specify MDISK.

### **WAIT code**

causes a disabled wait state PSW to be loaded after shutdown. The code in the PSW is a hexadecimal number between 1 and FFFF.

### **NOCKPT**

will suppress taking a checkpoint and load a disabled PSW of 9027 after the following message is displayed:

```
HCP9027W  SYSTEM TERMINATION COMPLETE WITHOUT CHECKPOINT
```

### **CANCEL**

causes a scheduled shutdown to be terminated. Any guests that received termination signals when the original SHUTDOWN command was issued continue to process those signals.

## Options

### **MDisk *userid vdev***

specifies the owning user ID and virtual device number of the minidisk that contains the CP module.

### **EXTent *n***

specifies which minidisk extent on a CP-formatted volume contains the module to be loaded. The variable *n* is a single decimal number in the range 1 - 9. 1 refers to the first extent formatted as "PARM", while 9 refers to the ninth such extent.

### **MOdule *filename***

specifies the file name of the module that SHUTDOWN is to load. The file type of the module must be MODULE.

### **OFFset *nnnnnnnn***

specifies the offset from the beginning of the disk (identified by *rdev*) at which the CMS-formatted minidisk that contains the module resides. You can specify up to 8 digits for OFFSET. For CKD and ECKD DASD, the offset is specified as the number of cylinders from the beginning of the disk. For FBA DASD, the offset is specified as the number of blocks from the beginning of the disk.

### **ORigin *hexloc***

specifies the address at which SHUTDOWN loads the module. This option is ignored by SHUTDOWN if the module that is being loaded was not generated with the RLDSAVE option. In this case, the module is loaded into storage at the location at which the module was generated.

When loading CP, the ORIGIN option is accepted, but CP relocates itself to location X'2000'.

### **Iplparms**

specifies information that is to be passed to the loaded program by SHUTDOWN.

#### ***text/?***

specifies information that is to be passed to the loaded program by SHUTDOWN. The information replaces all currently specified IPL parameters.

**Note:** When you enter *text* on the command line, IPLPARMS must be the last option that you specify. The format of the IPL parameters depends entirely on the options that are supported by the loaded program. For a list of IPL parameters supported by CP, see [Passing IPL Parameters in z/VM: System Operation](#).

#### **?**

causes SHUTDOWN to prompt the operator for up to three lines of data to be read from the console. A null entry can be used to signify that all parameters are entered.

SHUTDOWN command processing accepts a "?" or a "? " (a "?" followed by a blank) as valid arguments to initiate the prompting sequence for the IPL parameters. All other arguments are assumed to be IPL parameters and do not initiate the prompting sequence.

## Usage Notes

1. Because SHUTDOWN stops all virtual machine operations, consider warning all users of an impending shutdown as early as possible so that they can quiesce their systems in an orderly manner.

You can use the SET LOGMSG, MESSAGE, and WARNING commands to inform the affected users.

**Note:** CMS file pool servers shut down automatically without operator intervention unless a server is explicitly disabled from receiving a shutdown signal from CP. For more information, see [Startup Parameter Descriptions in z/VM: CMS File Pool Planning, Administration, and Operation](#).

2. Specifying SHUTDOWN with no operands causes the system to come to an orderly termination. When the termination completes, CP enters a disabled wait state.
3. SHUTDOWN terminates all outstanding Concurrent Copy (CONCOPY) sessions, and disbands the path groups of all DASD if CP is running in a native machine.
4. If REIPL is specified but *rdev* is omitted, the current system residence device is re-IPLed.

## SHUTDOWN

5. When REIPL is specified, CP attempts to do an automatic warm start. If the automatic warm start is to succeed:
  - a. It must have its warm start cylinders and checkpoint cylinders defined in the same location that the current system does.
  - b. The new CP module must contain the same versions of control blocks that are used by the checkpoint or warm start process, such as HCPCKIBK, HCPCKPBK, HCPPFXPG, HCPSYSCM, HCPSPBKB, HCPSPFBK, and HCPSPMBK.
  - c. The new CP module must contain the same version of the checkpoint (HCPCKP or HCPCKS) and warm start (HCPWRM or HCPWRS) modules.
  - d. Any CP system volumes that are shared between the old and the new system must have the same position in the CP\_Owned statement in the system configuration file.

Otherwise, the automatic warm start might fail and a hardware IPL with a FORCE start must be completed.

6. When REIPL is specified, CP uses the parm disk location from the previous IPL. If the parm disk was moved by using CPFMTXA or ICKDSF, CP might fail to find the CPLOAD module and issue message HCP6739E. Or CP might find the old CPLOAD module on the previous parm disk. Use the EXTENT option of the SHUTDOWN command to force CP to find the parm disk at its new location, or use the OFFSET option to specify the new location of the parm disk. It is recommended to move the parm disk only shortly before a planned REIPL.
7. When a SHUTDOWN is issued, the trace data for all active DEFERIO traces are not written.
8. All IPL parameters that are used with the IPLPARMS option remain in effect until they are changed with a subsequent SHUTDOWN REIPL command that specifies the IPLPARMS parameter, or the CP nucleus is shut down, or a SET IPLPARMS is issued.
9. If the REIPL option is used, some I/O subsystem features might not be available if the system that is brought up on the re-IPL supports a superset of the features that are supported by the system that was shut down. A full I/O reset is necessary to support all the features. A full I/O reset can be completed only by using the Load function on the processor's hardware console. For example, the Parallel Access Volumes feature of an IBM DASD Subsystem might not be available on re-IPL if the system that was shut down did not support that feature.
10. A user is enabled for signals only when a guest operating system is running in the virtual machine and is enabled to receive these signals.
11. If the WITHIN, BY, or IMMEDIATE operand is not specified on the SHUTDOWN command, and the system-default shutdown-signal timeout interval is greater than 0, the following occurs:
  - CP sends a shutdown signal to enabled users.
  - The start of the CP shutdown is delayed. The delay gives guest systems an opportunity to do their own shutdown processing. The CP shutdown is delayed until the shutdown-signal timeout interval elapses or all signaled guests shut down, whichever occurs first.

If the WITHIN, BY, or IMMEDIATE operand is not specified on the SHUTDOWN command and the system-default shutdown-signal timeout interval is 0, shutdown signals are suppressed and the CP shutdown begins immediately.

Use caution when you specify the WITHIN, BY, or IMMEDIATE operand on the SHUTDOWN command. The resulting interval might not provide guests with enough time to complete an orderly shutdown.

The WITHIN, BY, and IMMEDIATE operands specify the time interval for the entire z/VM system shutdown. The shutdown-signal timeout interval, which provides time for guests to shut down, is the difference between the entire z/VM system shutdown interval and the CP shutdown interval. When you specify the WITHIN or BY operand, the shutdown-signal timeout interval might not provide guests with enough time to complete an orderly shutdown. When you specify the IMMEDIATE operand, the shutdown-signal timeout interval does not provide guests with enough time to complete an orderly shutdown.



If the interval that you specify with the WITHIN or BY operand is equal to the CP shutdown interval, shutdown signals are suppressed and the CP shutdown begins immediately.

For examples of the interaction between the SHUTDOWN command timing operands, the CP shutdown interval, and the system-default shutdown-signal timeout interval, see [“Examples” on page 1794](#).

**Tip:**

- The system-default shutdown-signal timeout interval is specified by the SET SIGNAL SHUTDOWNTIME system configuration statement or command. For more information, see [“SET SIGNAL” on page 1701](#).
  - The CP shutdown interval is specified by the SET SHUTDOWNTIME system configuration statement or command. For more information, see [“SET SHUTDOWNTIME” on page 1698](#).
12. A user might issue the SET SHUTSIGNAL OFF command to temporarily make shutdown signals unavailable. The QUERY SIGNALS command can be used to display the signal status of a specified user or all users.
  13. After the SHUTDOWN command is initiated successfully, this message is sent to the primary system operator:

```
HCP2116I SHUTDOWN initiated at yyyy-mm-dd hh:mm:ss by userid
```

14. After the SHUTDOWN CANCEL command is issued, this message is sent to the primary system operator:

```
HCP2117I SHUTDOWN CANCEL issued at yyyy-mm-dd hh:mm:ss by userid
```

15. Users who receive the shutdown signal and need more information about the system shutdown can issue QUERY SHUTDOWN STATUS to get information about the current shutdown.

## Responses

### Response 1:

```
System shutdown may be delayed for up to interval seconds
```

is displayed when the command is accepted and some guests require time to shut down. If a user other than the primary system operator issues the SHUTDOWN command, this response is sent to both the command issuer and the primary system operator.

### Response 2:

```
SYSTEM SHUTDOWN STARTED
```

is displayed when the command is accepted and termination processing is in progress. If a Class A user other than the primary system operator issues the SHUTDOWN command, this response is sent to both the command issuer and the primary system operator.

### Response 3:

```
STARTING SYSTEM SHUTDOWN STEP name - description
```

is displayed for each step of the shutdown process as it proceeds. Each message identifies the *name* of the step and a *description* of its function. On a 3270 terminal, these messages are displayed on the first line of the screen, always showing the last step that was initiated. On the system console (SYSC) device, the messages are displayed one after another in line mode.

## Examples

1. In the following example, a guest is not provided their time to process signals because the duration is not long enough.

```
SET SHUTDOWNTIME 70
SET SIGNAL SHUTDOWNtime 30
SHUTDOWN WITHIN 90
```

Guests get only 20 seconds to respond to shutdown signals because CP requires 70 of the 90 seconds for CP shutdown.

2. In this example, assume that the time is exactly 11:58. The SHUTDOWN command's BY operand specifies a time of 12:00, which provides 120 seconds to process a shutdown.

```
SET SHUTDOWNTIME 70
SET SIGNAL SHUTDOWNtime 30
SHUTDOWN BY 12:00
```

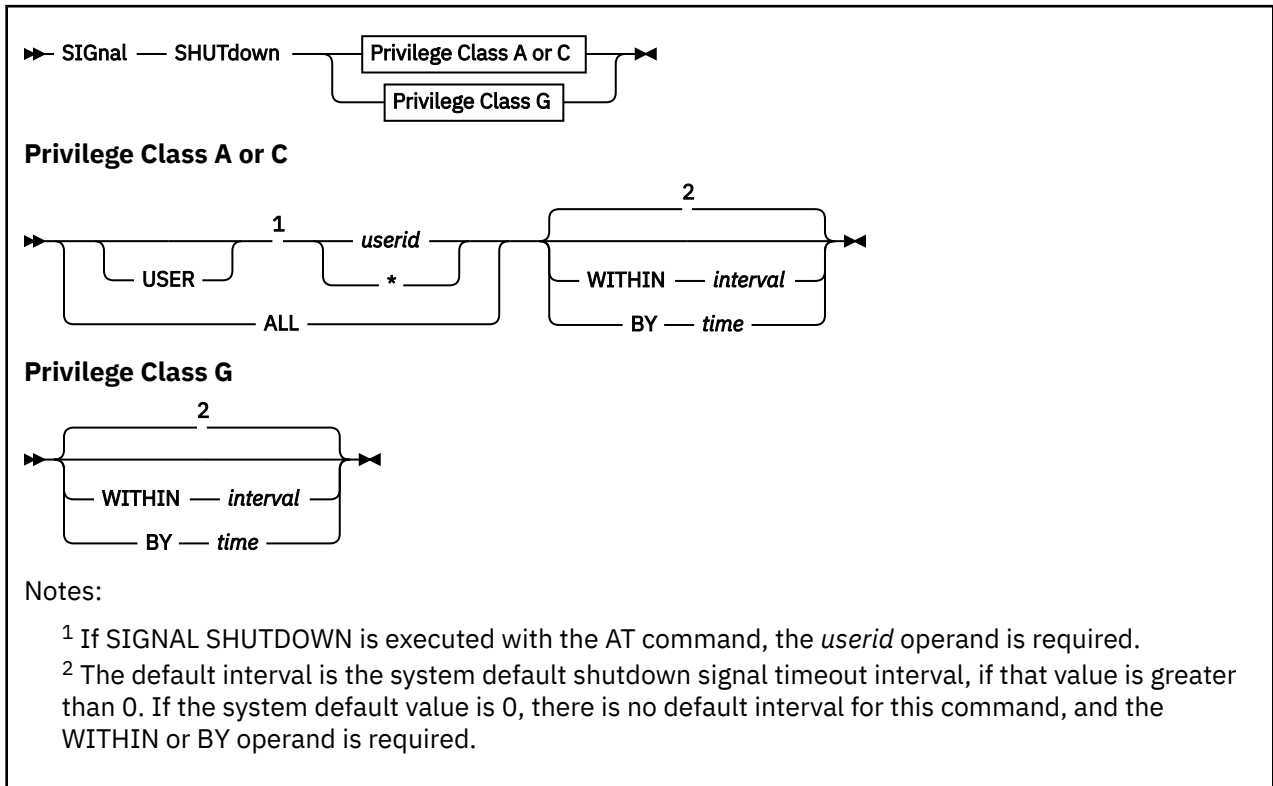
Guests get 50 seconds to respond to shutdown signals because CP needs only 70 of the 120 seconds for the CP shutdown.

## Messages

- HCP002E Invalid operand - *operand*
- HCP005E Invalid hexdata - *operand*
- HCP006E Invalid device type - *rdev*
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E DASD *rdev* offline
- HCP052E Error in CP directory
- HCP053E *userid* not in CP directory
- HCP126E DASD *rdev* error reading valid
- HCP128E DASD *rdev* error reading allocation record
- HCP160E *hexloc* exceeds the storage size
- HCP160E Hexloc *hexloc* exceed storage
- HCP959I *sysname* SYSTEM TERMINATION IN PROGRESS ON *yyyy-mm-dd*
- HCP961W SYSTEM SHUTDOWN COMPLETE FOR *sysname* ON *yyyy-mm-dd*
- HCP962I VM SHUTDOWN COMPLETED IN *duration* SEC
- HCP965E The IPLPARMS option is not allowed when this command is executed remotely with the AT command.
- HCP2116I SHUTDOWN initiated at *yyyy-mm-dd hh:mm:ss* by *userid*
- HCP2117I SHUTDOWN CANCEL issued at *yyyy-mm-dd hh:mm:ss* by *userid*
- HCP6014I *command* cancelled at user request.
- HCP6015E *command* cancelled because of console read error.
- HCP6016E System shutdown is already in progress
- HCP6017E System shutdown is not in progress
- HCP6019E Guests may not have time to shut down because VM SHUTDOWN requires *duration* seconds
- HCP6020E Guests do not have time to shut down because VM SHUTDOWN requires *duration* seconds
- HCP6021E Required SYSTEM operand missing on SHUTDOWN command; command rejected
- HCP6022E Incorrect value specified for SYSTEM - *sysname*
- HCP6028E Scheduled system shutdown has already begun.

- HCP6425I {CP TRACE TABLE RECORDING|TRACEID *traceid*} DEFERIO TRACE DATA WILL NOT BE WRITTEN.
- HCP6704E Missing token at end of line
- HCP6723E Requested parm disk not found in allocation map.
- HCP6739E *filename* MODULE not found
- HCP6881E *userid vdev* is not in CP directory.
- HCP6882E *userid vdev* is not a minidisk.
- HCP6883E *userid vdev* not accessible -- volume *valid* not mounted.
- HCP6884E Unable to access re-IPL minidisk because no free access slot is available.
- HCP6884E Unable to access re-IPL minidisk because of an I/O error.
- HCP6884E Unable to access re-IPL minidisk because the disk is not CMS formatted.
- HCP9015W SYSTEM TERMINATION FAILURE; MACHINE MALFUNCTION
- HCP9022W SYSTEM LOAD FAILURE; FATAL I/O ERROR
- HCP9023W SYSTEM TERMINATION FAILURE; MULTIPLE INCIDENTS
- HCP9025W SYSTEM TERMINATION COMPLETE, ENTERING WAIT STATE
- HCP9027W SYSTEM TERMINATION COMPLETE WITHOUT CHECKPOINT
- HCP9035E LOAD ORIGIN address of *hexloc* is not valid.
- HCP9275I SYSTEM TERMINATION IN PROGRESS; RESTART IGNORED
- HCP9276W SYSTEM TERMINATION FAILURE; PROGRAM ERROR
- HCP9277I SYSTEM TERMINATION COMPLETE, ATTEMPTING RESTART`yyyy-mm-dd` SYSTEM *sysname*

## SIGNAL



### Authorization

Privilege Class: A, C, G

### Purpose

Use `SIGNAL` to send a signal to a user or to all users who are enabled to receive signals.

A class A or C user can send a signal to a specific user or to all users who are enabled to receive signals, and log the users off after they have processed the signal.

A class G user can send a signal to the user's own virtual machine, if the virtual machine is enabled for signals.

### Operands

#### SHUTDOWN

indicates that a termination signal is to be sent.

#### userid

identifies the user to be signaled. Use the keyword `USER` when you specify a user ID of "all".

#### ALL

indicates that all users who are enabled to receive termination signals should be signaled.

#### WITHIN interval

sends the user a shutdown signal that times out when the specified interval elapses. The *interval* specified is a number of seconds in the range 1 - 32767.

#### BY time

sends the user a shutdown signal that times out when the designated time is reached. The *time* can be specified as *hh:mm* or *hh:mm:ss*. The equivalent interval in seconds must be in the range 1 - 32767 seconds.

## Usage Notes

1. A user is enabled for signals only when a guest operating system is running in the virtual machine and has enabled to receive these signals.
2. When the signal times out or the user machine indicates that it has terminated operation, whichever occurs first, the virtual machine is logged off if it has been disconnected.
3. If the WITHIN or BY operand is not specified, and the system default shutdown signal timeout interval is set to a value greater than 0 (by the SET SIGNAL SHUTDOWNTIME system configuration statement or command), that value is used as the signal timeout interval. If the system default value is 0, you must specify an interval on the SIGNAL command. For additional information about this behavior, see [“SET SIGNAL” on page 1701](#).
4. A user might have issued the SET SHUTSIGNAL OFF command to temporarily disable for receiving shutdown signals. The QUERY SIGNALS command can be used to display the signal status of a specified user or all users.
5. When the SIGNAL SHUTDOWN \* command or the class G SIGNAL SHUTDOWN command is issued, no message is sent to the primary system operator. When a SIGNAL SHUTDOWN is initiated successfully by a class A or C user with the SIGNAL SHUTDOWN ALL command or the SIGNAL SHUTDOWN *user* command, this message is sent to the primary system operator:

```
HCP2118I SIGNAL SHUTDOWN {ALL | user} issued at yyyy-mm-dd hh:mm:ss by userid
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP007E Invalid userid - *userid*
- HCP013E Conflicting option - *option*
- HCP045E *userid* not logged on
- HCP2110E Virtual machine is not enabled for signals
- HCP2110E User *userid* is not enabled for signals
- HCP2111E Virtual machine has not responded to a previous signal
- HCP2111E User *userid* has not responded to a previous signal
- HCP2112I Signal timeout interval has expired
- HCP2112I User *userid* signal timeout interval has expired
- HCP2113I User *userid* has reported successful termination
- HCP2118I SIGNAL SHUTDOWN {ALL | *user*} issued at *yyyy-mm-dd hh:mm:ss* by *userid*
- HCP6704E Missing token at end of line
- HCP6706E Invalid string - *string*
- HCP6706E Invalid time - *time*

## SILENTLY

---

```
►► SILENTly — command_string ◄◄
```

### Authorization

Privilege Class: Any (See usage note “1” on page 1798).

### Purpose

Use SILENTLY to execute another CP command, suppressing its responses if that command is so enabled. A command is enabled for response suppression by specifying the SILENT option on the DEFINE COMMAND / CMD or MODIFY COMMAND / CMD command or system configuration statement.

**Note:** Response suppression is supported only for the ATTACH, DETACH, and GIVE commands.

### Operands

#### *command\_string*

is the full text of the CP command to be executed. If the command is enabled for response suppression, command responses are suppressed. Warning, error, and informational messages produced by the command are not affected. If the command is not enabled for response suppression, it executes as if it was issued directly.

### Usage Notes

1. You must have the proper privilege class for the CP command to be executed.

### Examples

1. To attach device 1234 to user MVS and suppress the messages sent to the issuer, to the user who is the object of the ATTACH, and to the system operator:
  - a. Enable the ATTACH command for response suppression:

```
modify command attach silent
```

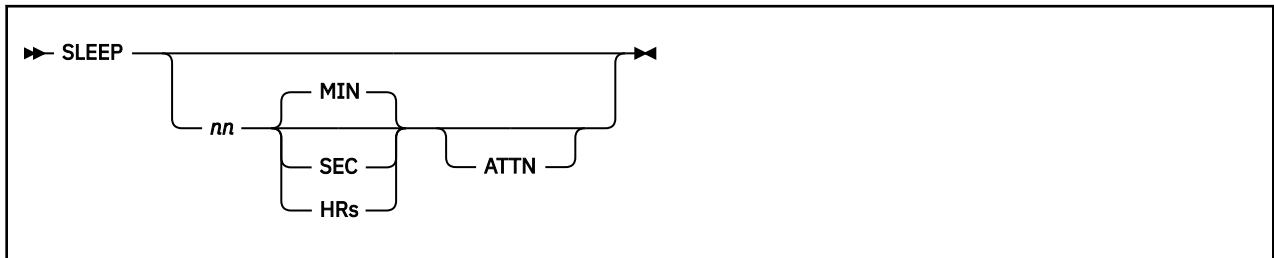
- b. Issue the SILENTLY command for ATTACH:

```
silently attach 1234 to mvs as 1234
```

### Messages

- HCP026E Operand missing or invalid
- HCP6704E Missing token at end of line

# SLEEP



## Authorization

Privilege Class: Any

## Purpose

Use SLEEP to stop your virtual machine for a specified period of time. When the specified period elapses, your virtual machine automatically resumes processing.

## Operands

***nn* SEC**

***nn* MIN**

***nn* HRs**

identifies the number of seconds, minutes, or hours, measured by the time-of-day clock, that your virtual machine is to remain stopped. The value *nn* can be any decimal number from 00 through 99. If you specify no time unit, CP assumes that *nn* indicates minutes.

### ATTN

specifies that if the SLEEP command with a time interval is issued through a DIAGNOSE X'08' instruction, the return code indicates whether the virtual machine restarted because the time interval expired (RC=0), or because an attention was signaled (RC=1457).

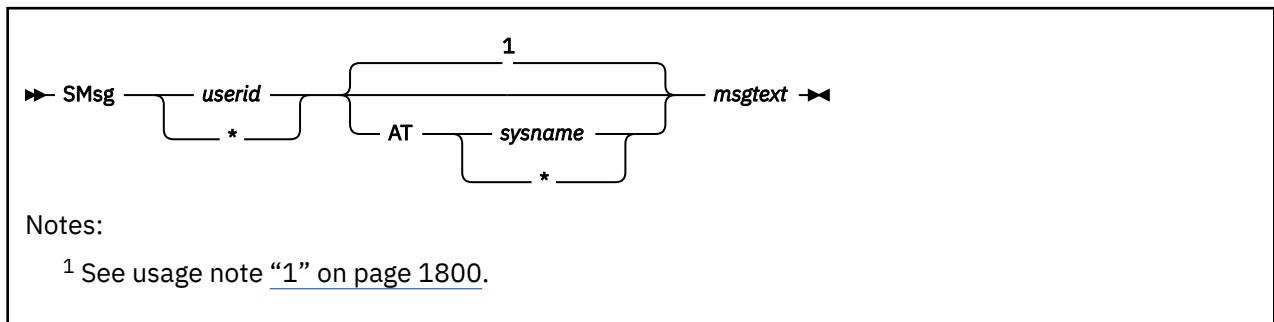
## Usage Notes

1. During the dormant period, the virtual machine does not run, but connection time is still counted.
2. Messages continue to be displayed even though your virtual machine is in the dormant state.
3. You can start your virtual machine at any time by pressing the ENTER key. If you specify a sleep interval in the command line, your virtual machine awakens automatically when the specified interval has elapsed.
4. If you do not specify an interval, your virtual machine remains dormant until you press the ENTER key.
5. If you enter the SLEEP command from a CP environment, you will return to the CP environment when the time interval has elapsed, or when you press the ENTER key.
6. If you enter the SLEEP command while in virtual machine environment, the end of the time interval or pressing ENTER returns your display to virtual machine mode without entering the CP environment.
7. The SLEEP command, with the time interval, is a convenient way to delay or schedule the execution of certain jobs that could be run more efficiently at a later time (for example, second shift).

## Messages

- HCP002E Invalid operand - *operand*

## SMSG



### Authorization

Privilege Class: G

### Purpose

Use SMSG to send a special message to a virtual machine programmed to accept and process the message.

### Operands

#### *userid*

\*

is the virtual machine to which the special message is to be sent. To send the message to yourself, specify an asterisk (\*).

#### **AT** *sysname*

sends the special message to a specific system in the SSI cluster to which this system belongs. The return code is sent to the command issuer.

#### **AT** \*

sends the special message only to the system where the command is issued. This is the default if the system is not a member of an SSI cluster.

#### *msgtext*

is the text of the message you are sending. The length of the message is limited by the number of characters remaining on the input line after the command and the appropriate operand are entered. Mixed DBCS data may be used as input message text.

### Usage Notes

- If the AT operand is not specified:
  - If the sender's system is not a member of an SSI cluster, the default is AT \*.
  - If the sender's system is a member of an SSI cluster, the message is sent to all members of the cluster, except in the following cases, when it is sent only to the local system:
    - If the specified user ID is OPERATOR.
    - If the specified user ID is logged on to the same system as the sender.
    - If the specified user ID or the sender is defined by a multiconfiguration virtual machine definition.
- Special messages do not appear on the receiving user's console because the messages are expected to be instructions or commands that are processed by the receiving virtual machine.
- Virtual machines can receive special messages only under the following conditions:



- The receiving virtual machine has issued the SET SMSG ON command, or set the VMCP SMSG flag at authorization time.
- The user wishing to receive special messages has issued an AUTHORIZE with the Virtual Machine Communication Facility (VMCF).

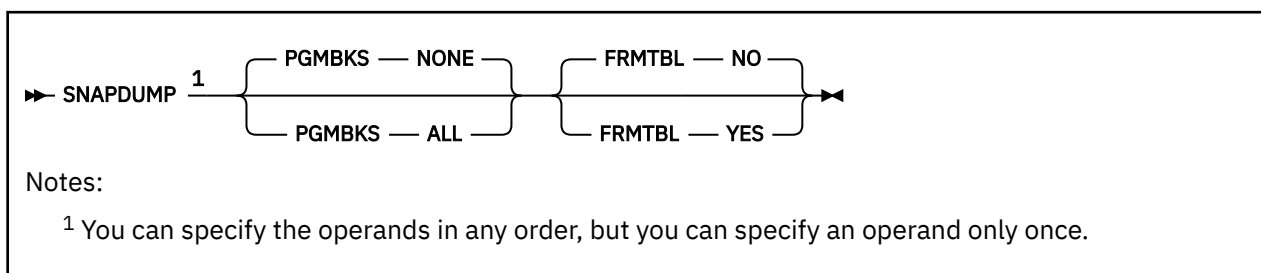
You can also receive special messages through IUCV by issuing the SET SMSG IUCV command.

4. If an external security manager (ESM) is installed on your system, you might not be authorized to enter this command. However, messages sent to or from the system operator are not subject to authorization checking by the ESM. For additional information, contact your security administrator.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP057I *userid* not receiving; *text*
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.
- HCP6600E An error was detected by installation-wide {*CP module name*|exit point 1210} - return code *nnnn*

## SNAPDUMP



### Authorization

Privilege Class: A

### Purpose

Use SNAPDUMP to generate a system dump without terminating and restarting the system. The dump that is created is identical to a hard abend dump, except that it will not contain PGMBKs unless they were requested.

### Operands

#### FRMTBL NO

#### FRMTBL YES

specifies whether the full CP frame table is included in the dump. If you specify NO, the default, the frame table is not included and the dump includes a correlation table describing only the frames whose contents were dumped. YES forces inclusion of the frame table. IBM recommends using the default unless IBM support personnel instruct you to do otherwise in order to diagnose a specific problem.

#### PGMBKS NONE

#### PGMBKS ALL

specifies whether user PGMBKs are included in the dump. If you specify NONE, the default, no user PGMBKs are included. ALL forces inclusion of all PGMBKs that aren't paged out. IBM recommends using the default unless IBM support personnel instruct you to do otherwise in order to diagnose a specific problem.

### Usage Notes

1. This command is particularly useful when diagnosing a hung user situation. The settings specified by the SET DUMP command determine where the output will go.
2. The frame table and PGMBKs together constitute the vast majority of CP-owned frames, because they contain entries describing each 4K unit of real and virtual memory, respectively. The CP frame table is 1/128th the size of configured memory (8 GB per TB). PGMBKs take up 8K for each 1MB of virtual memory; though they are pageable, a PGMBK must remain resident as long as any page of that megabyte is resident. Omitting these structures from dumps when they are not likely to be needed for diagnosis dramatically reduces dump size and dumping time. When debugging problems using a snap dump, it is very rare to need information from the frame table or any of the PGMBKs. You will be directed by IBM support personnel to include these items if this will aid in problem determination.
3. Because SNAPDUMP command operands can influence the size of the dump, CP allocates dump space corresponding to the maximum possible size of the dump. Make sure you have enough DASD space available to support another dump after this one is taken.
4. Although this command will generate a dump without taking your system down, all activity in the system is stopped. If you specify FRMTBL YES or PGMBLKS ALL, the amount of time necessary to

dump the system could be sufficient to cause your communication lines to drop, which could adversely affect your system.

Consider this before using this command:

- You may want to send a message to all logged-on users warning them of the impending quiesce (similar to what an operator might do before issuing SHUTDOWN). Although the system should not go into a shutdown, it will appear to the users that the system has stopped while the dump is being taken.

In a Multi-VSwitch LAG environment, if the LPAR issuing the SNAPDUMP is also the controlling LPAR for all the LAG ports, this can also cause network outages on the other LPARs in that LAG group.

5. The abend code in a dump created from use of the SNAPDUMP command will be SNP001.

## Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP9265I The SNAPDUMP command has been entered causing the non-destructive dump function to be called. The system will be quiesced until the dump is complete.
- HCP9662I The SNAPDUMP command has been issued, however the current dump setting is OFF. No dump will be generated.
- HCP9267E The system is unable to determine the dump device settings following a non-destructive dump. The dump device is now set to OFF. To assign a different device, enter the SET DUMP command.
- HCP1858I The non-destructive dump function has completed.

## SPACE

►► SPACe — rdev ◄◄

### Authorization

Privilege Class: D

### Purpose

Use SPACE to override real printer file carriage control commands and force single-spaced output.

### Operands

#### *rdev*

identifies the printer whose output file is to be single spaced.

### Responses

Response 1:

```
PRT rdev SPACING   userid   FILE spoolid RECS {nnnn} COPY [*] nnn
                               {nnnK}
                               {nnnM}
SEQ sss CL c typ
```

#### *rdev*

identifies the printer.

#### *userid*

identifies the user whose file was active and single spaced.

#### FILE *spoolid*

identifies file currently being processed.

#### RECS *nnnn*

#### RECS *nnnK*

#### RECS *nnnM*

identifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

#### COPY [\*] *nnn*

identifies the remaining number of copies of the file to be created. The asterisk (\*) indicates that the 3800 printer is to copy each page *nnn* times before going on to the next page.

#### SEQ *sss*

identifies the sequence number for the file on the printer.

#### CL *c*

is the spool file class.

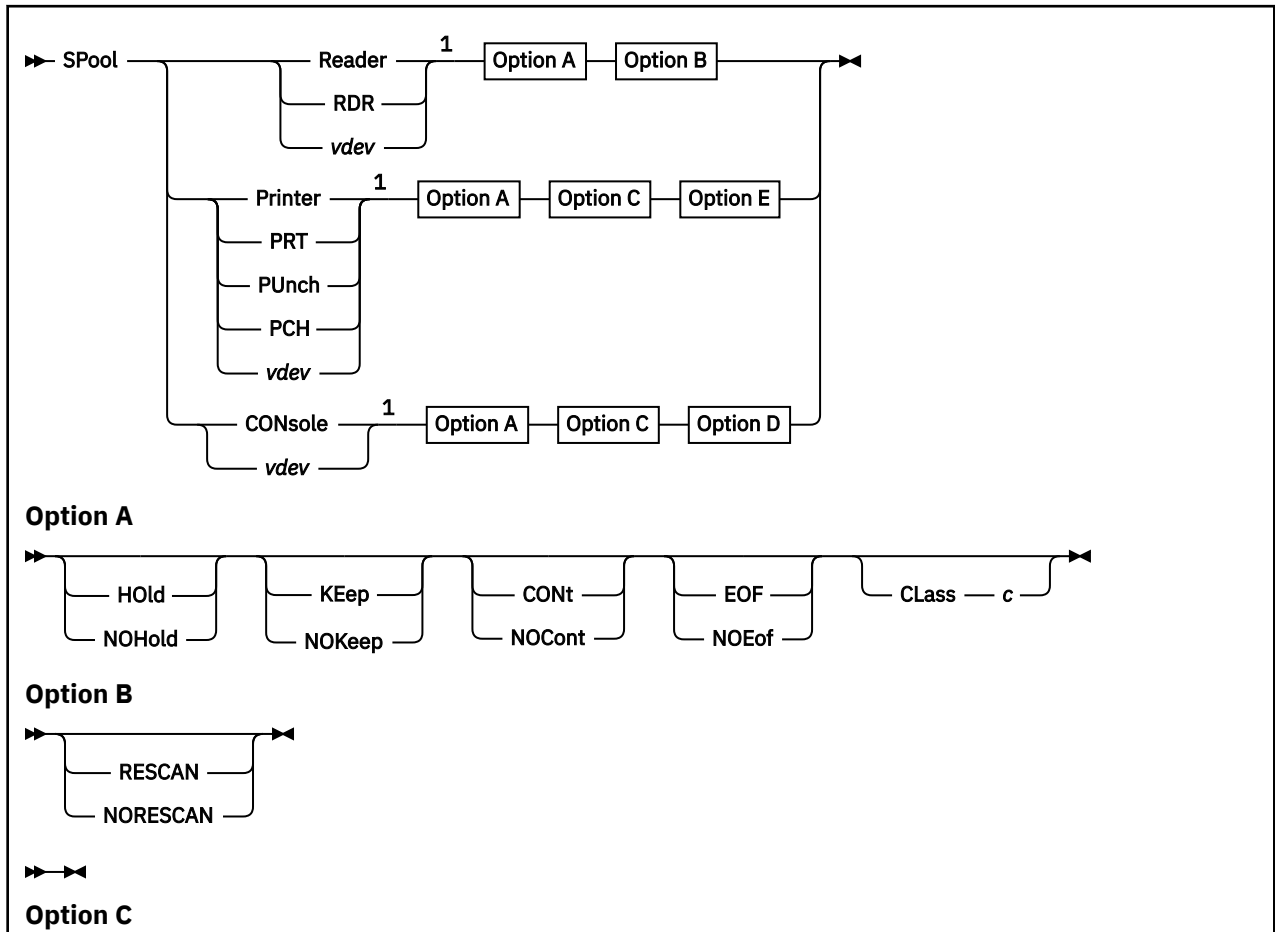
#### *typ*

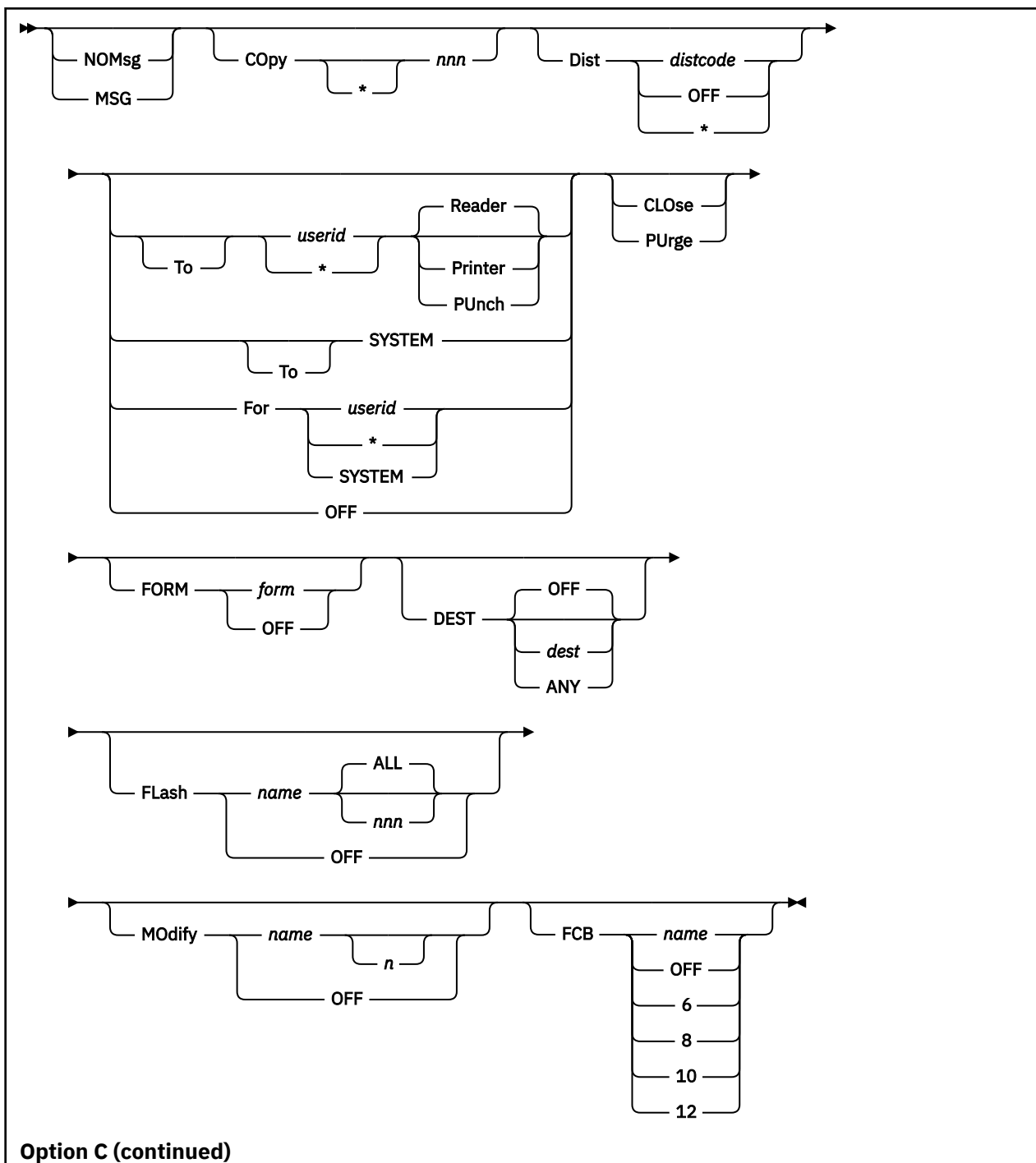
shows device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP, or SYS.

## Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP021E A real device number was not supplied or it is invalid.
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP046E *type rdev* offline
- HCP140E *type {rdev/ldev}* attached to *user id*
- HCP141E *type rdev* not active
- HCP417E *{rdev/vdev}* is not an output spooling device.
- HCP418E *{rdev/vdev}* is not a spooled device.
- HCP419E *{rdev/vdev}* is not a spooled printer.

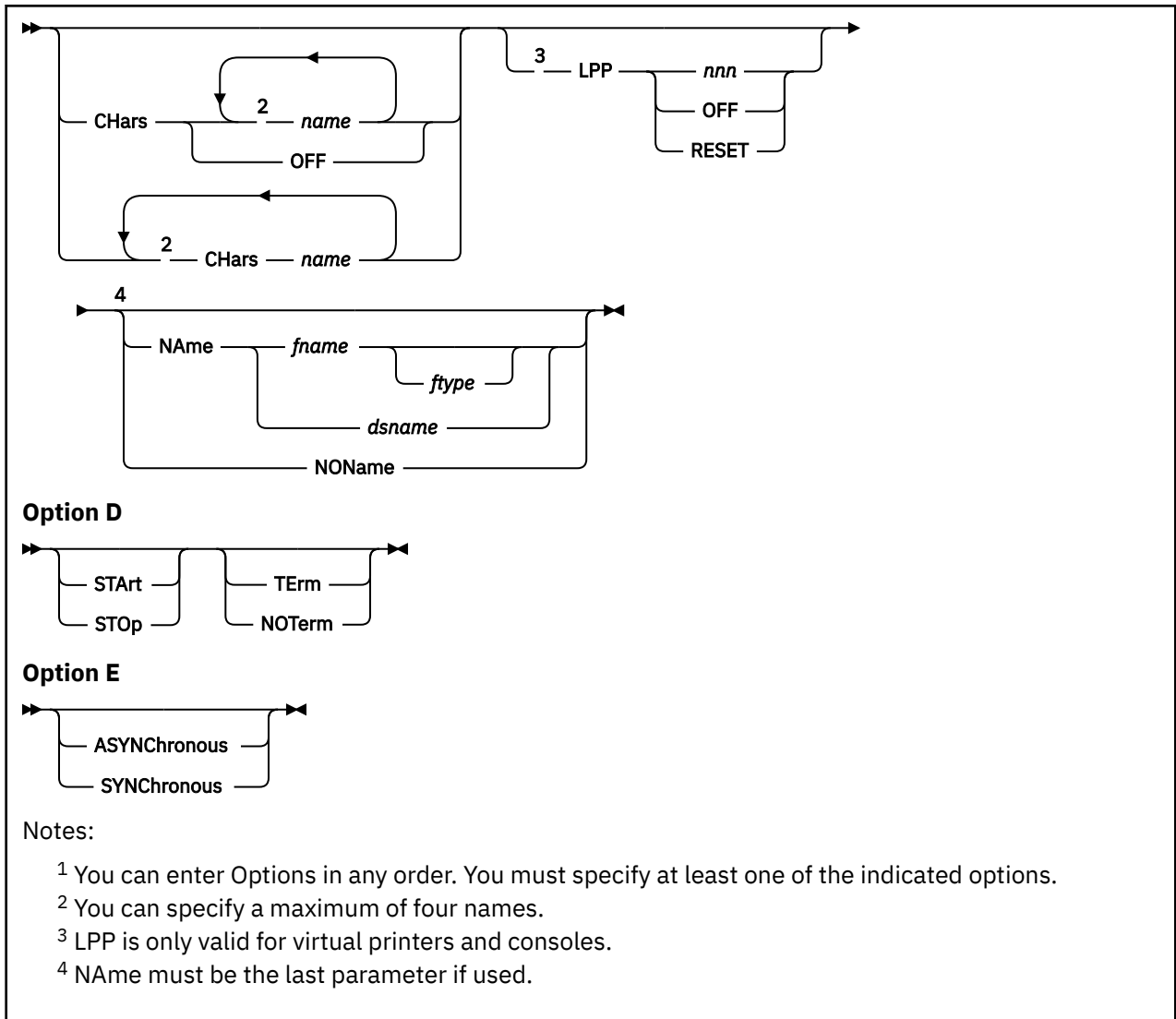
# SPOOL





Option C (continued)

# SPOOL



## Authorization

Privilege Class: G

## Purpose

Use SPOOL to set control options for one or more of your virtual spool devices. The options you select modify operational functions associated with your virtual reader, printer, punch, or console. These options also control the disposition of files after they have been processed.



## Operands

### Reader

**RDR**

### Printer

**PRT**

### PUnch

**PCH**

### CONsole

*vdev*

modifies the options for all of your virtual readers, printers, punches, or console spooling devices, including the starting or stopping of spooling operations. Alternatively, you can enter the virtual device number *vdev* of the virtual spooling device whose options you want to modify.

## Options

### HOLD

#### NOHOLD

indicates whether you want to HOLD files.

If you specify HOLD for an output device, all the files created by the device are placed in user hold status. When a spool file is in user hold status, it is not processed at its destination device (the real printer, punch, or virtual reader) until you remove the hold status by issuing the CHANGE command with the NOHOLD option.

If you specify HOLD for an input device, files are retained after they have been read. Unless you specify the KEEP operand for the file or for the device, the file remains in the reader queue and is the next file eligible for processing. This is in contrast to the hold status on a spool file, which makes the file ineligible for processing.

A reader may not be spooled with the string CONT/RESCAN/HOLD/NOKEEP, as this would cause the same file to be processed repeatedly. If a SPOOL command is issued which would result in the reader being spooled with this combination, the reader will be spooled KEEP to remove the conflict.

Specify NOHOLD to allow files to be purged after processing.

### KEep

#### NOKeep

indicates whether you want to retain a spool file after it has been processed. Specify KEEP if you want to retain files after they have been processed.

For a virtual printer or punch directed to a real device, after the file has been printed or punched, it is placed in user, hold status, while retaining its keep status.

For a reader, KEEP also places your spool file in user, hold status after it has been read, but keep status is not set on the file. Note that the reader DEVICE hold option does not affect file hold status.

A reader may not be spooled with the string CONT/RESCAN/HOLD/NOKEEP, as this would cause the same file to be processed repeatedly. If a SPOOL command is issued which would result in the reader being spooled with this combination, the reader will be spooled KEEP to remove the conflict.

Specify NOKEEP to reset the keep status so that files are not retained in the system after processing has been completed.

### CONT

#### NOCont

controls the continuous spooling option.

For output devices, CONT causes all CLOSE requests to be ignored unless specified with EOF override option. NOCONT resets the option.

For readers, CONT specifies that reading will continue without end-of-file indications at the end of each virtual spool file. Reading continues until the virtual reader queue has been scanned once from

beginning to end. After each spool file is read, scanning is continued at the next logical file (unless RESCAN is specified).

NOCONT specifies that end-of-file is signaled after each spool file is read. The reader resets at the next logical file.

**RESCAN****NORESCAN**

When a reader is spooled CONT, RESCAN causes the first logical file in the virtual reader queue to always be the next file read. This allows files which have been ordered or changed during continuous processing to become eligible for processing.

A reader may not be spooled with the string CONT/RESCAN/HOLD/NOKEEP, as this would cause the same file to be processed repeatedly. If a SPOOL command is issued which would result in the reader being spooled with this combination, the reader will be spooled KEEP to remove the conflict.

NORESCAN causes scanning to resume at the next logical file in the virtual reader queue.

**EOF****NOEOF**

For a printer or punch, EOF causes the device to be closed automatically when 50000 records are placed into the active spool file. Punches are closed immediately at 50000 records. If page controls are found in a print file within the next 100 records, the file is closed at the end of the page. If no page controls are found by then, it is presumed there are none, and the file is closed after 50100 records.

For a reader, the EOF/NOEOF option specifies whether end-of-file is reached (as defined by the CONT/NOCONT option). EOF specifies that a unit exception is reflected on the first read past the end of the file. Further reads result in a normal read of the next file, if any, or with a unit check if the device is empty. NOEOF specifies that unit check is reflected on the first read past end-of-file, or any subsequent read to an empty reader.

**Class c**

specifies the spool class of the device. The operand *c* is a 1-character alphanumeric field whose values can be A through Z, 0 through 9, or asterisk (\*). The CLASS option determines which destination devices can process the spool file. If you print or punch a file from one of your virtual devices, the file is assigned the class that is associated with the device. The file is then printed or punched on a real device that also has this class associated with it. The operator sets this class.

The class associated with a virtual reader determines which files it recognizes for processing. If you specify a specific class, the reader processes only files of the same class. If you specify asterisk (\*), the reader processes all files, regardless of class.

**NOMsg****MSG**

indicates whether you want to be informed about file printing and punching. If you specify MSG, you are informed of file printing and punching if the file is printed or punched on the same system as your user ID. The message is sent to you when the file begins printing or punching on a real device. Specify NOMSG if you do not want messages sent to you when the file begins printing or punching on a real device.

**Copy [\*] nnn**

identifies the number of copies to be printed or punched when the file is spooled to the real unit record device. Note that regardless of what number you specify for COPY, only one spool file is actually created. However, z/VM continues to associate the COPY number with the spool file until the spool file is processed by a real printer or punch. At that time, z/VM produces the specified number of real printed or punched copies. If the real printer is a 3800 device, an asterisk (\*) indicates that the printer is to print each page *nnn* times before going on to the next page, where *nnn* is a decimal number from 1 to 255.

**Dist *distcode*****Dist OFF****Dist \***

specifies the distribution code to be assigned to spool files created on this virtual device. The operand *distcode* is one to eight characters long. You can override *distcode* using the DIST option on the CLOSE or CHANGE commands.

DIST OFF and DIST \* reset the distribution code of the spool files to the DIST code in the system directory at the time you logged on and will not reset to an updated value until logoff.

**TO *userid*****TO \*****TO SYSTEM**

sends the output of the virtual device to the virtual reader, virtual printer, or virtual punch of the specified user ID. The distribution code on the spool files retains the distribution code specified for the originator in the system directory.

When you specify the keyword TO, you retain control of the spool file. For example, you can use the TRANSFER command to retrieve the file from the specified user ID.

TO SYSTEM is equivalent to specifying OFF and resets the transferred spool option.

**Reader****Printer****PUnch**

identifies the spool file queue to which your file is directed. If you do not specify a queue, the file is transferred to the other user's reader queue.

**CLOse**

closes the spool file on the specified device regardless of the CONT setting for the device. CLOSE is equivalent to entering CLOSE with the EOF option. If PURGE is specified, CLOSE is not necessary.

**PURge**

closes and purges the spool file on the specified device, regardless of the CONT setting for the device. If PURGE is specified, the CLOSE option is not necessary. PURGE is equivalent to entering CLOSE with the PURGE option.

**FOR *userid*****FOR \*****FOR SYSTEM**

except for console output, the user ID option transfers the spool file from the virtual device to the corresponding virtual device of the specified user ID. z/VM transfers console output to the receiver's virtual printer. The distribution code on the spool files becomes the distribution code specified for *userid* in the system directory.

You can specify FOR \* to change the spool file ownership and destination code to those of your own virtual machine. Specifying FOR \* is equivalent to specifying the OFF operand for the device.

FOR SYSTEM is equivalent to specifying OFF and resets the transferred spool option. If you specify FOR *userid*, you cannot specify either the TO or the OFF operand.

**OFF**

resets the TO or FOR option from the previous spooling command.

**FORM *form***

specifies the form name for the resulting spool file or files. The operand *form* is one to eight characters long. The FORM option determines which destination devices can process the spool file. If you print or punch a file from one of your virtual devices, the file is assigned the form associated with the device. The file is then printed or punched on a real device that also has this form associated with it. The operator sets this form.

**FORM OFF**

resets the form to the system default as specified by the installation.

**DEST *dest***

specifies the destination value for the resulting spool file or files. The operand *dest* is a 1- to 8-character value. The DEST option determines which destination devices can process the spool file. If you print or punch a file from one of your virtual devices, the file is assigned the destination associated with the device. The file is then printed or punched on a real device that also has this destination associated with it. The operator sets this destination.

**DEST ANY**

indicates that the resulting spool file or files can be processed on any CP output device that meets other selection criteria, regardless of the device's DEST setting.

**DEST OFF**

indicates that the resulting spool files are processed by a device specifically started, or they are defaulted to OFF. OFF is the initial setting of the destination value for virtual devices.

**Flash ALL****Flash *nnn***

signifies that a form overlay contained in the 3800 printer is to be superimposed (or flashed) onto the specified number of copies of the output. The operand *name*, 1- to 4-characters long, identifies the forms overlay to be used in the projection. The operand *nnn* must be in the range from 0 to 255. The number specifies that the first *nnn* copies of output are to be flashed.

For example, if you specify COPY 4 and FLASH *name* 2, the first two copies of your file are flashed, while the last two copies remain unflashed. If ALL is specified, the flash count matches the copy count. ALL is the default.

**Flash OFF**

resets the flash parameters. Blanks are inserted into the flash name field of the specified spool file or files, and the count is set to 0.

**MODify *name n***

assigns a copy modification module to be used when printing the spool file or files on the 3800. This function makes it possible to alter text by either preventing the printing of certain information or adding labels to output. The name must be one to four characters long.

The operand *n* selects one of the character sets specified by the CHARS operand, to be used for the copy modification text. The values 0, 1, 2, and 3 correspond to the order in which the table names have been specified. If no *n* is specified, the first character set (0) is assumed.

**MODify OFF**

resets the modify parameters. Blanks are inserted into the modify name field of the spool file or files and, when printed, no modification occurs. The modify number *n* is set to 0.

**FCB *name*****FCB 6****FCB 8****FCB 10****FCB 12**

specifies the forms control buffer to be used when printing the spool file or files on the 3800 printer. The operand *name* is the 1- to 4-character name of an FCB image or the number 6, 8, 10, or 12. This number tells the 3800 to print the entire spool file at 6, 8, 10, or 12 lines per inch, regardless of the size of the paper that is currently loaded. An FCB specified at 10 will print only on a 3800 model 3 printer.

**FCB OFF**

resets the FCB parameter. Blanks are inserted into the FCB name field of the spool file or files and, when printed, a default FCB will control the vertical spacing of the output text.

**CHars *name0 name1...*****CHars *name0 CHars name1...***

specifies the name of the character set or sets used when printing the spool file or files on the 3800 printer. Each name must be from one to four characters long, with a maximum of four names specified. The multiple use of CHARS need only be used if *namen* would conflict with an option name

(for example, FORM) or with a user ID. If more than one name is specified, along with modification, the order in which the names appear determines which table is used for copy modification text.

### **CHars OFF**

resets the CHARS parameters. Blanks are inserted into the name fields of character sets used for output generation. The resulting spool file or files, therefore, uses a default character set when actually printed on a 3800 printer.

### **LPP *nnn***

### **LPP OFF**

### **LPP RESET**

sets the lines per page value for a virtual printer or a virtual console. '*nnn*' must be a decimal number with a range of 30 to 255.

If 'OFF' is specified, internal default values will be used for lines per page. The defaults are the same as those which have been used on prior releases of VM.

'RESET' will reset the lines per page value for the virtual printer or console to the global system value. This is also the default lines per page setting for virtual printers and consoles when they are defined.

The global system value for LPP may be defined in the system configuration file. If it is not defined, the default global value for the system is 'OFF'. This causes all virtual printers and consoles to be defined with a default LPP value of 'OFF'. See the *z/VM: CP Planning and Administration* for information about the [USER\\_DEFAULTS Statement](#).

The lines per page value is used by VM functions which generate virtual printer and console output. For CP, these include console spooling, CP dump and trace output, DDR, load maps generated by HCPLDR, and SPXTAPE log files. CMS functions which use the lines per page value for virtual printer output include the PRINT, ASSEMBLE, UPDATE, GENMSG, TAPE, FILEPOOL FORMAT AUDIT, and SET DOSLNCNT commands.

The lines per page value for a virtual printer is displayed in the response for QUERY VIRTUAL PRT command (as well as other QUERY commands which display virtual printer status).

The lines per page value for a virtual console is displayed in the response for QUERY VIRTUAL CONS command (as well as other QUERY commands which display virtual console status).

The lines per page value for a virtual printer or console may be obtained by application programs by issuing DIAGNOSE X'260' subcode 08.

### **NAME *fname ftype***

### **NAME *dsname***

specifies the file name and optional file type, or the data set name that will be given to each spool file the virtual device creates. The *fname* and *ftype* values must be eight characters or fewer. The data set name may be up to 24 characters; however, if the name has more than 16 characters, it is truncated to 16 characters.

### **NOName**

resets the NAME option. Blanks are inserted into the name field for spool files the virtual device creates.

### **STArt**

begins placing input and output from your display station into a spool file. Full-screen data output operations are not placed in the spool file. After the console is closed, the file becomes a printer spool file. START applies only to consoles.

### **STOp**

terminates the spooling of display station input and output. STOP does not close the console spool file. STOP applies only to consoles.

### **TErm**

displays the display station input and output on your display screen and places it in a spool file. The TERM operand has no effect unless the START function is also in effect. TERM applies only to consoles.

## NOTerm

suppresses the display of input and output on your display screen while placing it in a spool file. The display of console input and output is not suppressed if:

- It is the result of CP commands entered from CP mode
- The output line immediately precedes a virtual machine read to the console.

The NOTERM operand has no effect unless the START function is in effect. NOTERM applies only to consoles.

Full screen mode applications such as XEDIT and full screen CMS are not supported when the console is spooled NOTERM.

## ASYNChronous

indicates you want to process other commands while the SPOOL command completes.

## SYNChronous

prevents you from executing other commands until the SPOOL command completes.

## Usage Notes

1. When specifying [TO] *userid*, if you omit the keyword TO, *userid* cannot be T or TO. If you are transferring your output to your own virtual machine, you may specify *userid* as an asterisk (\*). If you specify TO *userid*, you cannot specify either the OFF or FOR operand.
2. Full screen functions may hang a user whose console is spooled NOTERM. For example, if a user is in XEDIT and tries to QUIT, the keyboard remains locked.
3. SET IMSG OFF suppresses informational messages for this command.
4. A spool file is created when the first record for a printer or punch is processed by the virtual unit record simulation routines in CP. CP associates this spool file with the virtual unit record device and assigns it a number called the *spoolid* (spool file identification). You can use *spoolid* to identify the file that you have produced, and to manipulate the file using the z/VM CP spooling commands, such as ORDER, CHANGE, and PURGE.

When you print or punch a file, CP displays its *spoolid*. You can use the *spoolid* to find out various kinds of information about the file using the QUERY command.

5. It is important to remember that the SPOOL command controls both operational characteristics of your virtual spooling devices and the disposition of files after they have been processed.

For example, you use the SPOOL command to select the files you want to process on your virtual readers. The SPOOL READER command does not change any of the characteristics associated with the files in your reader queue, but simply indicates whether a file is purged or retained after being read, and if it is retained, whether the file remains eligible for immediate reprocessing.

The files in your reader queue are controlled through a combination of file characteristics and the options in effect for the virtual reader on which they are processed. Input files do not derive their characteristics from the reader, except for hold status, which is set as a function of the KEEP option. [Table 41 on page 1815](#) summarizes the disposition of a reader file after it has been processed, depending on the hold/keep settings specified for the reader, and the keep setting of the file.

For output devices (virtual printers, punches, and consoles) use the SPOOL command to define the characteristics of the spool files you are creating. You can use the SPOOL command to indicate whether files are to be held from further processing, and whether they are to be retained after they have been processed. This information is associated with the spool file itself, regardless of the characteristics of the device on which the file is to be processed. In addition, use SPOOL to direct the output files that you are creating to unit record devices (readers, printers, and punches) associated with other virtual machines.

Table 41. Disposition of Reader Files Depending upon HOLD/KEEP Settings

<b>File Status</b>	<b>Device Options</b>	<b>Disposition</b>
NOKEEP	NOHOLD/NOKEEP	The file is purged.
NOKEEP	NOHOLD/KEEP	The file is retained in user hold status. KEEP is not set on the file.
NOKEEP	HOLD/NOKEEP	The file is retained and remains eligible for processing on your virtual reader.
NOKEEP	HOLD/KEEP	The file is retained in user hold status. KEEP is not set on the file.
KEEP	Any setting	The file is retained in user hold status. KEEP status remains on the file.

All of your output files derive their characteristics from the virtual device on which they were created. If the output device is spooled with the KEEP and NOHOLD options, the file is printed or punched, and then placed in HOLD status, retaining its KEEP characteristics. If you change the file status to NOHOLD, the file is processed at the real device one more time and once again is placed in HOLD status. If you do not want to retain the file, you must change both its NOHOLD status and NOKEEP characteristic, or purge it using the PURGE command.

6. Unless you reset these values, the following options are the default values for your spool files:

<b>Device</b>	<b>Default Options</b>
Reader	NOHOLD NOKEEP NOCONT NORESCAN EOF CLASS A <sup>1</sup>
Punch	NOHOLD NOKEEP NOCONT CLASS A <sup>1</sup> OFF COPY 001 NOMSG NONAME NOEOF DEST OFF
Printer	NOHOLD NOKEEP NOCONT CLASS A <sup>1</sup> OFF COPY 001 NOMSG NONAME NOEOF DEST OFF
Console	NOHOLD NOKEEP NOCONT CLASS T <sup>1</sup> OFF COPY 001 TERM TO * PRINTER NOMSG STOP NONAME NOEOF DEST OFF

<sup>1</sup>These are default classes only if a specific class is not part of your entry in the system directory.

7. You can transfer a spool file to another z/VM user by issuing the SPOOL, CHANGE, CLOSE, or TRANSFER commands. Use the CHANGE and TRANSFER commands to transfer already existing files. Use the SPOOL or CLOSE commands to transfer files that you are in the process of creating, or files you plan to create. For example, if you want to transfer the printer or punch files you are creating to another user's virtual reader, enter **spool vdev to userid rdr**, where *vdev* identifies by number or type your virtual machine output device, and *userid* identifies the virtual machine of the receiving user. Designating a reader, a printer, or a punch is optional; the default designation is reader. The command defines the user queue to which you are directing the file.

If you want to print or punch a file for someone else, enter **spool vdev for userid**, where *vdev* identifies by number or type your virtual machine output device, and *userid* identifies the virtual machine of the receiving user.

8. When a file changes ownership, as when it is changed to or from another user, the *spoolid* number changes also; the new number is obtained from a pool of numbers available to the file recipient.

Thus, if user SAM transfers his file number 3 to user HARRY, he gets the response:

```
RDR 0003 SENT TO HARRY RDR AS 0018
```

Harry now has the file as file number 18.

9. If you specify options that apply to a 3800 printer, such as COPY(\*), FLASH, MODIFY, FCB, or CHARS, the 3800 printer must be controlled by CP rather than the print server.

10. If DEST is not specified on a command that sets a DEST value, the current value is unchanged.
11. If an external security manager is installed on your system, you may not be authorized to transfer spool files using the *TO* and *FOR* options. For additional information, contact your security administrator.
12. The SPOOL command may not be used to direct files to the RDR queue of a user with a password of NOLOG. SPOOL may be used to direct files to the PRT and PUN queues of a user with a password of NOLOG.

## Responses

Response 1:

```
{RDR} FILE spd1 SENT {TO } userid {RDR } AS spd2 RECS {nnnn}
{PRT}                {FROM}      {PRT }
{PUN}                {PUN }      {nnnK}
{CON}                {CON }      {nnnM}
```

Continued:

```
COPY [*]nnn c {HOLD } {KEEP }
              {NOHOLD} {NOKEEP}
```

### RDR PRT PUN CON

identifies the spool file queue. If you are transferring a file to another user's virtual machine, this field indicates the queue from which the file originated. If you are receiving a file from another user, this field indicates the file queue to which the file has been added.

### *spd1*

identifies the spool file at the time the command is issued.

### *TO userid* *FROM*

*TO* indicates the user to whom the file has been sent. *FROM* indicates the user sending the file.

### RDR PRT PUN

identifies the spool file queue. If you are transferring a file to another user's virtual machine, this field indicates the queue to which the file has been added. If you are receiving a file from another user, this field indicates the file queue in which the file originated.

### *spd2*

identifies the file after the command is complete.

### RECS *nnnn* RECS *nnnK* RECS *nnnM*

identifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

### COPY [\*] *nnn*

identifies the number of copies of the file you want printed or punched when the file is produced. The asterisk (\*) indicates that the 3800 printer is to copy each page *nnn* times before going on to the next page, where *nnn* is a decimal number from 1 to 255. If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored and duplication occurs normally.

### *c*

identifies the spool file class assigned to the file.



**HOLD**  
**NOHOLD**

identifies the user hold status associated with the file.

**KEEP**  
**NOKEEP**

identifies the keep status associated with the file.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP030E Copies missing or invalid
- HCP035E Device type missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP052E Error in CP directory
- HCP053E *userid/value* not in CP directory
- HCP088E An invalid number was supplied for {COPY|FLASH} - *operand*.
- HCP418E {*rdev/vdev*} is not a spooled device.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP424I The requested option(s) would result in incompatible settings for reader *vdev*. The reader has been spooled KEEP to remove this conflict.
- HCP429E *type vdev* spool error; file {held|purged}
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*.
- HCP1551E *value* is an invalid number for MODIFY.
- HCP1552E More than {4|8} operands were specified for the {CHARS|CLASS} option.
- HCP1559E The DSNAME is truncated to 16 characters.

## SPXTAPE

---

### Authorization

Privilege Class: D, E, G

### Purpose

Use SPXTAPE to:

- Save standard spool files and system data files on tape
- Restore SPXTAPE-format files from tape to the spooling system.

Standard spool files are:

- Printer spool files
- Punch spool files
- Reader spool files.

System data files are:

- Image libraries
- National language support files, such as message repository files
- Named saved systems
- Saved segments
- System trace files
- User class restructure files.

SPXTAPE provides the following major functions:

#### **CANCEL**

Immediately cancels an SPXTAPE DUMP, LOAD, or SCAN operation. See [“SPXTAPE CANCEL” on page 1834.](#)

#### **DUMP**

Reads files from the spooling system and writes them on tape. See [“SPXTAPE DUMP” on page 1836.](#)

#### **END**

Ends an SPXTAPE DUMP, LOAD, or SCAN operation at the completion of the current file. See [“SPXTAPE END” on page 1844.](#)

#### **LOAD**

Reads SPXTAPE-format files from tape and writes them into the spooling system. See [“SPXTAPE LOAD” on page 1846.](#)

#### **SCAN**

Determines what files exist on tape in SPXTAPE format. See [“SPXTAPE SCAN” on page 1857.](#)

The spooling operator can use SPXTAPE to selectively store on tape and retrieve standard spool files and system data files. The scan and selective-read-in functions allow the operator to avoid an overload in the spooling system. Should an overload occur, SPXTAPE allows the operator to quickly store the files temporarily on tape.

The general user can use SPXTAPE to save on tape and retrieve the standard spool files and system trace files that belong to that user ID.

**Note:** The status of an outstanding SPXTAPE operation can be determined using the QUERY VIRTUAL command. See the description of QUERY VIRTUAL TAPES for more details.

Log Files

The activities of each SPXTAPE command are recorded in at least two spool files: a command summary log and one or more volume logs. These logs are created by the SPXTAPE DUMP, LOAD, and SCAN commands.

Both logs have the same routing and disposition as the virtual console of the user who enters the SPXTAPE command (or causes the SPXTAPE operation to begin). If you enter the command `spool cons *` before you enter the SPXTAPE command, all logs are sent to your virtual reader. If you want to change the spooling of your console, you should do it before you enter the SPXTAPE command, or the logs may be sent to different destinations.

#### *Command Summary Log:*

The command summary log records the progress and status of an SPXTAPE DUMP, LOAD, or SCAN operation. Most command responses, as well as any error messages, are recorded in this log. A new command summary log is created for each SPXTAPE DUMP, LOAD, or SCAN command, unless the command is appended to a previous SPXTAPE command.

Each command summary log has a unique name. The file name and file type have the following format:

```
aabbcccc hhmmss
```

#### **aa**

is the month the SPXTAPE command is issued.

#### **bb**

is the day the SPXTAPE command is issued.

#### **cccc**

is the SPXTAPE operation: DUMP, LOAD, or SCAN.

#### **hh**

is the hour the SPXTAPE command is issued.

#### **mm**

is the minute the SPXTAPE command is issued.

#### **ss**

is the second the SPXTAPE command is issued.

The command summary log contains a header (response 1) for the command that creates the log. If the command that creates the log is issued with the APPEND operand, a header is also written in the log for each appended command.

#### *Volume Log:*

The volume log records information about the files processed by an SPXTAPE DUMP, LOAD, or SCAN command that are associated with a particular tape volume. There is one volume log for each tape volume for each SPXTAPE command. For example, if you use the SPXTAPE DUMP command with the LEAVE operand 10 times to save files on a single tape volume, 10 volume logs are created. If you use a single SPXTAPE DUMP command to dump files to 10 different tape volumes (and all 10 are actually used), 10 volume logs are created.

The volume log is a record of what is dumped to, loaded from, or scanned on a tape volume. For an SPXTAPE LOAD operation, the volume log may record more than what is actually loaded from the tape. A failure or operator action that occurs after a file has been read from the tape and logged (such as spool space being exhausted, or the SPXTAPE CANCEL command being used against the tape drive) may prevent the file from being stored in the spooling system. The files that are not loaded successfully are reported with messages sent to the SPXTAPE user and recorded in the command summary log.

Each volume log has a unique name. The file name and file type have the following format:

```
aabbccvvv hhmmssstt
```

#### **aa**

is the month the SPXTAPE command is issued.

**bb**

is the day the SPXTAPE command is issued.

**c**

is the first letter of the SPXTAPE operation: D = DUMP, L = LOAD, S = SCAN.

**vvv**

are the three least significant digits of the virtual device number of the tape drive.

**hh**

is the hour the SPXTAPE command is issued.

**mm**

is the minute the SPXTAPE command is issued.

**ss**

is the second the SPXTAPE command is issued.

**tt**

are the two least significant digits of the volume sequence number, which is the number of this volume in the sequence of volumes processed so far on this device for this command.

The volume log contains a header (response 1) for the command that creates the log. If the command that creates the log is issued with the APPEND operand, a header is also written in the log for each appended command. Each header is followed by the file information (response 2 entries) for the files selected by the command identified in the header.

## Usage Notes

1. SPXTAPE cannot load or scan files dumped by the SPTAPE command, which is no longer supported. See *z/VM: Migration Guide*.
2. SPXTAPE supports all tape drives fully supported by z/VM. For a list of supported devices, see *z/VM: General Information*.
3. On tapes that have standard tape labels, SPXTAPE skips over the header label group. If the header label group was written on a drive with a format that is incompatible with the drive where the tape is currently mounted, SPXTAPE DUMP cannot read the label and will write over it.
4. SET IMSG OFF suppresses the display of SPXTAPE informational messages, error message HCP1909E, and response 9. However, SET IMSG OFF does not suppress the request to mount a new tape when the current one is full. Message HCP1916I is written in the command summary log only and not displayed, even if IMSG is set ON.
5. Specifying a range of one virtual device number on an SPXTAPE command is the same as specifying just one device. For example, `spxtape dump 181-181` is the same as `spxtape dump 181`.
6. When an SPXTAPE command is issued with a device number range, the command begins processing if a supported tape drive is attached to at least one device number in the range. Unsupported devices and device numbers where no device is attached are ignored and are reported only if there is no valid device in the range. On SPXTAPE DUMP, tape devices attached read-only are ignored and are reported only if no device is attached write-enabled.  
  
If the range includes any device already active for a matching SPXTAPE DUMP, LOAD, or SCAN command, the active command range is expanded to include the drives covered by the expanded range. This capability can be used to add drives at any time for performance or to recover from tape drive errors. However, if the previous command was not entered with the APPEND operand, the command to expand the range cannot include any file selection criteria.
7. If all the tape drives on an active SPXTAPE command are reset (using RESET, DETACH, GIVE, or some other command), or if the issuer of the SPXTAPE command logs off or is forced off, the effect is the same as issuing the SPXTAPE CANCEL command for each tape drive.
8. You may specify selection criteria operands in any order. All selection criteria specified on a command must be met for a file to be selected.
9. A user with only class E command privileges cannot process reader, printer, or punch files.

A class G user can process only files on the user's own reader, printer, punch, and system trace file queues, or files on tape that were dumped from these queues.

A class G user should not be given access to tapes containing files that belong to other users. Although SPXTAPE does not permit the class G user to load or scan a file that belongs to another user, there are other commands and programs the user could use to read the contents of the tape.

10. **Attention:** Using multiple SPXTAPE commands to load and dump files simultaneously is not recommended. Simultaneous SPXTAPE operations may overlap. For example, if a DUMP PURGE request matches the selection criteria of a file that was just loaded by a LOAD request, the file might be dumped and purged from the system before it can be used.

If multiple SPXTAPE DUMP commands are issued simultaneously for the same files, some files might not be dumped as many times as specified in the commands. This is because SPXTAPE DUMP opens a file to be dumped, but a file already open (except a named saved system or saved segment file) cannot be dumped.

11. SPXTAPE attempts to dump or load the selected files in the minimum amount of time by overlapping and optimizing the DASD and tape I/O activity. To maximize performance in a DUMP operation, unless you know the files being dumped will occupy less than one volume, it is recommended that you use at least one tape drive for every 4-6 spool DASD volumes, with a minimum of two (to allow overlapping volume mounting).
12. Tape-handling options (disposition operands) of LEAVE, REWIND, and RUN (rewind and unload) are provided for the DUMP, LOAD, and SCAN operations. The disposition is processed when the command is complete (or when the operation is ended by an SPXTAPE END command) to position the tape for subsequent commands. For LOAD and SCAN, the disposition also determines what conditions cause completion of the command.

SPXTAPE rewinds and unloads the current tape and waits for another tape to be mounted when:

- DUMP encounters either the physical end-of-tape or a permanent tape I/O error.
- LOAD or SCAN with the RUN disposition encounters either the logical end-of-tape (the end of the tape file written by SPXTAPE DUMP) or a permanent I/O error.

If SPXTAPE END is issued for a tape drive that is currently active for an SPXTAPE command with the APPEND disposition, the disposition is changed to RUN; that is, the tape is rewound and unloaded.

If SPXTAPE CANCEL is issued for a tape drive, no disposition is processed. After the cancel has been performed (and response 8 issued), SPXTAPE does not perform any further I/O operations (including RUN or REWIND) to the canceled drive.

13. SPXTAPE supports the dumping of files that are too large to fit on a single tape reel or cartridge. A file whose size exceeds the capacity of a single tape volume is split across as many volumes as are required to contain it.

To load such a multivolume file, all volumes that contain parts of the file must be read by a single SPXTAPE command. The order in which the volumes are read does not matter as long as all the parts are included. SPXTAPE puts the parts together in the proper sequence.

14. While SPXTAPE is processing a file in the spooling system (except a named saved system or saved segment), the file can be queried but not manipulated in any way.
15. While an SPXTAPE operation is running on a virtual tape device, no other virtual I/O is permitted to that device. If other virtual I/O is attempted, it is rejected.
16. Almost all of the processor time used by an SPXTAPE command is scheduled and accounted for on the command issuer's user ID.

## Responses

Response 1:

When SPXTAPE DUMP, LOAD, or SCAN processes any files to or from a tape volume, or SPXTAPE with the APPEND disposition appends any files to be processed, the following header is written in the command summary log and the volume log:

```
date time userid command
```

**date**

is the date the command is issued.

**time**

is the time the command is issued.

**userid**

is the user ID of the user who enters the command.

**command**

is a string containing the first 132 characters of the SPXTAPE command.

Response 1 may include an additional line with the following format if the tape was positioned at the beginning of the volume and contained a standard tape label.

```
Tape Volume ID: valid
```

where *valid* is the 1 to 6 character tape volume identifier from the standard tape label.

Response 1 may also include the following line if encryption keys are specified on the ATTACH or SET RDEVICE commands.

```
Tape Volume Encryption: Key manager defaults are being used
```

Response 2:

When SPXTAPE DUMP, LOAD, or SCAN processes a file to or from a tape volume, the file is identified by the following entry in the volume log:

```
USERID FILE QUEUE FILENAME FILETYPE OPENDATE OPENTIME ORIGINID
userid file queue filename filetype opendate opentime originid

SIZE SEG_STAT HOLDSTAT RECCOUNT C FORM DEST LASTPAGE
size seg_stat holdstat reccount c form dest lastpage
```

**Note:** The heading is included before the first response 2 entry for that command in the log. It is repeated each time the flow of response 2 entries for the command is interrupted by some other response or message.

**userid**

is the user ID of the owner of the file. On LOAD, if the owner's user ID does not exist on the system or has reached the maximum number of spool IDs allowed for that virtual machine, this is the user ID that entered the command.

**file**

is the spool ID that identifies the file. On LOAD, this is the new spool ID assigned to the file on the system. If the file could not be loaded because no spool IDs are available, this field contains the spool ID of the file when it was dumped.

On LOAD or SCAN, if the file to be loaded or scanned is a duplicate of a file on the system (the SEG\_STAT field of this response contains the entry DUP\_FILE), this is the spool ID of the file on the system.

**queue**

is the source or destination queue (PRT, PUN, RDR, IMG, NLS, NSS, TRF, or UCR) that identifies the type of file being processed.

**filename**

is the file name of the file.

**filetype**

is the file type of the file.

**opendate**

is the date the file was created, in the format *mm/dd/yy*.

**opentime**

is the file creation timestamp, in the format *hh:mm:ss*.

**originid**

is the user ID of the creator of the file. On LOAD, if the file owner's user ID does not exist on the system or has reached the maximum number of spool IDs allowed for that virtual machine, the file owner's user ID is placed in this field.

**size**

is the number of spool data pages the file occupies.

**seg\_stat**

is the status of the file:

**PARTIAL**

indicates the volume contains only a part of the file, which is continued on one or more other volumes, and this part is *not* the one that completes the processing of the file.

**Note:** If a tape read error occurs on LOAD or SCAN, SPXTAPE cannot determine if the file is complete on this volume and puts PARTIAL in this field.

**COMPLETE**

indicates the dump, load, or scan of the file has been completed on this volume. Either the entire file is contained on this volume, or the part of the file on this volume is the one that completes the processing of the file.

**DUP\_FILE**

indicates the NODUP operand was specified on a LOAD or SCAN and the file on the tape duplicates a file already on the system:

- For a standard spool file or system trace file, this means all the attributes of the two files are identical.
- For a system data file other than a system trace file, this means the file name, file type, and class of the two files are identical. For a named saved system or saved segment, class A (active) and class R (restricted active) files with the same file name and file type are also considered to be duplicates.

The FILE field of this response contains the spool ID of the file on the system. The file on the tape is not loaded or scanned.

**NO\_LOAD**

indicates the file has not been loaded because:

- If the file is a standard spool file or a system trace file, one of the following conditions exists:
  - Both the user who owned the file when it was dumped and the user who entered the SPXTAPE LOAD command have run out of available spool IDs.
  - The user who owned the file when it was dumped does not exist on the system, and the user who entered the SPXTAPE LOAD command has run out of available spool IDs.
- If the file is a type IMG, NLS, NSS, or UCR system data file, the system has run out of available spool IDs.

**holdstat**

is the hold status of the file:

**USER**

indicates the file is user-held.

**SYS**

indicates the file is system-held.

**USYS**

indicates the file is both user-held and system-held.

**NONE**

indicates the file is not held. This is the entry for a system data file.

**reccount**

is the number of records in the file.

**c**

is the class of the file.

**form**

is the operator form name associated with the file. For a system data file, this field is blank.

**dest**

is the destination value of the file. For a system data file, this field is blank.

**lastpage**

is the highest relative page processed to or from the file.

Response 3:

When SPXTAPE DUMP, LOAD, or SCAN finishes processing, the following response is written in the command summary log and displayed on the screen.

```

SPXTAPE {DUMP} COMMAND {COMPLETED } ON VDEV {vdev } [WITH ERRORS]
        {LOAD}          {ENDED           } {vdev1-vdev2}
        {SCAN}          {CANCELED        }
                          {TERMINATED}

TIME STARTED:           hh:mm:ss
TIME ENDED:             hh:mm:ss

TAPE COUNT:            nnn
FILES PROCESSED:      nnn,nnn,nnn
SPOOL PAGES:          nnn,nnn,nnn

```

**Note:** If the number of Files or Spool Pages processed is larger than 999,999,999, it will continue to display as 999,999,999.

**DUMP****LOAD****SCAN**

is the SPXTAPE operation that has finished.

**COMPLETED**

indicates the operation completed on its own accord.

**ENDED**

indicates the user ended the operation with the SPXTAPE END command.

**CANCELED**

indicates the user canceled the operation with the SPXTAPE CANCEL command or by resetting the last tape drive.

**TERMINATED**

indicates the system terminated the operation. One or more messages are issued to identify the reason.

**VDEV vdev****VDEV vdev1-vdev2**

indicates the virtual device number, or the range of virtual device numbers, on which the SPXTAPE operation was active.

**WITH ERRORS**

indicates there was a problem processing at least one file. The specific files and problems are reported in messages and recorded in the logs.

**TIME STARTED: hh:mm:ss**

indicates the time the SPXTAPE command started.

**TIME ENDED: hh:mm:ss**

indicates the time the SPXTAPE command ended.

**TAPE COUNT: nnn**

indicates the number of tape volumes processed by the operation.



**FILES PROCESSED: *nnn,nnn,nnn***

indicates the number of files processed by the operation. A file that is split between two or more volumes is counted only once.

**SPOOL PAGES: *nnn,nnn,nnn***

indicates the number of spool pages processed by the operation. This includes XAB data pages and map pages. For SCAN, this is the number of pages that would be required to hold the files if they were loaded.

Response 4:

The following response is written in the command summary log and the volume log and displayed on the screen when:

- A user enters the SPXTAPE command with the END or CANCEL disposition selecting an active tape device
- SPXTAPE DUMP with the APPEND disposition has completed the processing of the selected files
- SPXTAPE processing is complete on a tape.

**Notes:**

1. For DUMP, this response is written at the beginning of the volume log; for LOAD or SCAN, it is written at the end.
2. If only one tape drive and one tape volume are being processed, this response is written only in the volume log and not displayed.

```

SPXTAPE {DUMP} {COMPLETED} } ON VDEV vdev [WITH ERRORS]
        {LOAD} {ENDED}       }
        {SCAN} {CANCELED}    }
                {WAITING}    }
                {TERMINATED} }

TAPE NUMBER:      vdev-nnn
FILES PROCESSED:  nnn,nnn,nnn
SPOOL PAGES:     nnn,nnn,nnn

```

**Note:** If the number of Files or Spool Pages processed is larger than 999,999,999, it will continue to display as 999,999,999.

**DUMP****LOAD****SCAN**

is the SPXTAPE operation that was active.

**COMPLETED**

indicates the operation completed on its own accord.

**ENDED**

indicates the user ended the operation with the SPXTAPE END command.

**CANCELED**

indicates the user canceled the operation by using the SPXTAPE CANCEL command or by resetting the last tape drive.

**WAITING**

indicates the tape drive is being processed by DUMP with the APPEND disposition and is waiting for another DUMP command selecting more files. When in this state, the tape is still "open" (incomplete, tape marks not written). To complete the tape, enter another SPXTAPE DUMP command with a disposition other than APPEND, or enter the SPXTAPE END command.

**TERMINATED**

indicates the system terminated the operation. One or more messages are issued to identify the reason.

**VDEV *vdev***

identifies the virtual device number of the tape drive on which the SPXTAPE operation is processing.

**WITH ERRORS**

indicates there was a problem processing at least one file. The specific files and problems are reported in messages and recorded in the logs.

**TAPE NUMBER: *vdev-*nnn****

identifies the virtual device number of the tape drive followed by the volume sequence number for this command (the number of tape volumes processed so far on this device for this command). The three least significant digits of the device number correspond to the last three characters of the file name of the volume log that records the processing done by this command on this tape volume on this device. The two least significant digits of the sequence number correspond to the last two characters of the file type of the volume log.

**FILES PROCESSED: *nnn,nnn,nnn***

indicates the number of files processed by the operation for this volume. If a file is split between two or more volumes, only the part that completes the processing of the file is counted (the one for which response 2 is issued with COMPLETE in the SEG\_STAT field).

**SPOOL PAGES: *nnn,nnn,nnn***

indicates the number of spool pages processed by the operation for this volume. This includes XAB data pages and map pages. For SCAN, this is the number of pages that would be required to hold the files if they were loaded.

**Note:** This number may differ from the actual pages contained on the volume if there are any partial files on the volume. For DUMP, XAB data pages and map pages are reported with the first volume that contains part of the file. For LOAD or SCAN, XAB data pages and map pages are reported with the volume that completes the file (the one for which response 2 is issued with COMPLETE in the SEG\_STAT field).

Response 5:

The following response is written in the command summary log and the volume log and displayed on the screen when the end of the tape is reached during:

- DUMP processing
- LOAD or SCAN processing with the RUN disposition.

```
SPXTAPE {DUMP} END-OF-TAPE ON VDEV vdev [WITH ERRORS];
        {LOAD}
        {SCAN}

MOUNT NEXT TAPE

TAPE NUMBER:      vdev-nnn
FILES PROCESSED:  nnn,nnn,nnn
SPOOL PAGES:      nnn,nnn,nnn
```

**Note:** If the number of Files or Spool Pages processed is larger than 999,999,999, it will continue to display as 999,999,999.

**DUMP**  
**LOAD**  
**SCAN**

is the SPXTAPE operation that was active.

**VDEV *vdev***

identifies the virtual device number of the tape drive on which the SPXTAPE operation is processing.

**WITH ERRORS**

indicates there was a problem processing at least one file. The specific files and problems are reported in messages and recorded in the logs.

**TAPE NUMBER: *vdev-*nnn****

identifies the virtual device number of the tape drive followed by the volume sequence number for this command (the number of tape volumes processed so far on this device for this command). The three least significant digits of the device number correspond to the last three characters of the file name of the volume log that records the processing done by this command on this tape volume on this

device. The two least significant digits of the sequence number correspond to the last two characters of the file type of the volume log.

**FILES PROCESSED: *nnn,nnn,nnn***

indicates the number of files processed by the operation for this volume. If a file is split between two or more volumes, only the part that completes the processing of the file is counted (the one for which response 2 is issued with COMPLETE in the SEG\_STAT field).

**SPOOL PAGES: *nnn,nnn,nnn***

indicates the number of spool pages processed by the operation for this volume. This includes XAB data pages and map pages. For SCAN, this is the number of pages that would be required to hold the files if they were loaded.

**Note:** This number may differ from the actual pages contained on the volume if there are any partial files on the volume. For DUMP, XAB data pages and map pages are reported with the first volume that contains part of the file. For LOAD or SCAN, XAB data pages and map pages are reported with the volume that completes the file (the one for which response 2 is issued with COMPLETE in the SEG\_STAT field).

When you receive this response, the tape drive enters the mount-pending state. The drive remains in this state until you mount and ready another tape. If this is a DUMP operation and there are other drives active that you believe will hold the rest of the files, do nothing. The mount-pending condition is removed when the command completes.

For the SPXTAPE operation to complete, you must mount enough volumes to complete any files in the process of being dumped, loaded, or scanned that span volumes. If a file takes more than one additional volume, you receive this response at the end of each volume until you reach the last volume.

If you do not want to continue the operation on this drive, do one of the following:

- Enter the SPXTAPE END command to remove the drive from the active drives or, if this is the last active drive, to end the operation.
- Enter the SPXTAPE CANCEL command for the device. However, if this was a DUMP operation, and if the device specified in the response is the last or only tape drive in the operation, using CANCEL could cause a file that was started on a previous tape volume to be not completed on this volume. You should use CANCEL only if you want to immediately halt the operation on the device regardless of the status of the files.

If all drives enter the mount-pending state and you take no action, the command remains dormant.

Response 6:

When SPXTAPE DUMP is dumping files to tape, it reports its progress at the interval requested with the PROGRESS\_interval option (or about every 15 seconds by default) by displaying the following response on the screen.

```
SPXTAPE DUMPING {vdev          }; nnn,nnn,nnn FILES, PAGES nnn,nnn,nnn
                 {vdev1-vdev2}
nn% COMPLETE
```

**Note:** If the number of Files or Spool Pages processed is larger than 999,999,999, it will continue to display as 999,999,999.

**DUMPING**

identifies the SPXTAPE operation that was active.

***vdev***

***vdev1-vdev2***

is the virtual device number or range on which the SPXTAPE operation is processing.

***nnn,nnn,nnn* FILES**

indicates the number of files processed to date by the operation. The rate at which this number changes is an inverse function of the size of the files being processed.

**PAGES *nnn,nnn,nnn***

indicates the number of spool pages processed to date by the operation. The rate at which this number changes is the rate at which data is being transferred to tape. Usually this is less than the transfer rate of the tape drives because in most cases performance is governed by the DASD read rates.

***nn%* COMPLETE**

indicates the percentage of the selected pages that have been dumped. If you are using the APPEND disposition and selecting additional files, this number can *decrease* as SPXTAPE includes the additional files in its processing.

If you do not receive this response during a DUMP operation, then SPXTAPE is not dumping any files. Possible reasons are:

- One or more tape drives are in "WAITING" state (identified by response 4 or 10), waiting for another SPXTAPE DUMP command.
- One or more tape drives are in mount-pending state (identified by response 5), waiting for additional tapes to be mounted.
- One or more tapes are in intervention-required state.
- You placed your virtual machine in CP READ state.

Response 7:

When SPXTAPE LOAD or SCAN is loading or scanning files, it reports its progress at the interval requested with the PROGRESS\_interval option (or about every 15 seconds by default) by displaying the following response on the screen.

```
SPXTAPE {LOADING } {vdev      }; nnn,nnn,nnn FILES, PAGES nnn,nnn,nnn
        {SCANNING} {vdev1-vdev2}
```

**Note:** If the number of Files or Spool Pages processed is larger than 999,999,999, it will continue to display as 999,999,999.

**LOADING  
SCANNING**

identifies the SPXTAPE operation that is active.

***vdev***

***vdev1-vdev2***

is the virtual device number or range on which the SPXTAPE operation is processing.

***nnn,nnn,nnn* FILES**

indicates the number of files processed to date by the operation. The rate at which this number changes is a direct function of the number of files selected and an inverse function of their size.

**PAGES *nnn,nnn,nnn***

indicates the number of spool pages processed to date by the operation. The rate at which this number changes is the rate at which data is being read from tape. It is a direct function of the percentage of files that are selected and their distribution on the tape. For LOAD, it may be less than the transfer rate of the tape drives because the load rate can be governed by the speed of the spooling system DASD.

If you do not receive this response during a LOAD or SCAN operation, then SPXTAPE is not reading the tape. Possible reasons are:

- One or more tape drives are in "WAITING" state (identified by response 4 or 10), waiting for another SPXTAPE LOAD or SCAN command.
- One or more tape drives are in mount-pending state (identified by response 5), waiting for additional tapes to be mounted.
- One or more tapes are in intervention-required state.
- You placed your virtual machine in CP READ state.

Response 8:

Each tape drive on which an SPXTAPE operation is initiated is reported with the following response written in the command summary log and displayed on the screen.

```
SPXTAPE {DUMP } INITIATED ON VDEV vdev
        {LOAD }
        {SCAN }
        {CANCEL}
        {END }

```

**DUMP**  
**LOAD**  
**SCAN**  
**CANCEL**  
**END**

is the SPXTAPE operation that was started.

***vdev***

is the virtual device number of the tape drive on which the SPXTAPE operation is running.

Response 9:

When a file is dumped with the PURGE operand, or when a file is loaded, the following response is sent to the file owner, if currently logged on. For DUMP, the response is issued after the tape is "closed" (trailers and tape marks have been successfully written), because files are purged in groups that correspond to a tape volume that has been successfully completed. For LOAD, the response is issued after each individual file has been placed on the queue.

```
{RDR} FILE spd1 SENT {TO } userid {RDR} AS spd2 RECS {nnnn}
{PRT}                {FROM} {PRT} {nnnK}
{PUN}                {PUN} {nnnM}
                    {**** PURGED **** }

CPY [*]nnn c {HOLD } {KEEP }
              {NOHOLD} {NOKEEP}

```

**RDR**  
**PRT**  
**PUN**

identifies the virtual device on which the file is being created or read.

***spd1***

is the spool ID of the file associated with the command user.

***userid***

is the user receiving the file.

**RDR**  
**PRT**  
**PUN**

identifies the spool file queue that now contains the file.

***spd2***

is the spool ID of the file associated with the user ID in the response.

**RECS *nnnn***

**RECS *nnnK***

**RECS *nnnM***

indicates the number of logical records in the file. If the number is greater than 9999, it is rounded to the nearest thousand and shown as *nnnK*, where *nnn* is the number of thousands. If the number is greater than 999499, it is rounded to the nearest million and shown as *nnnM*, where *nnn* is the number of millions.

**CPY [\*]*nnn***

indicates the number of file copies to be printed or punched when the file is produced. If the number is preceded by an asterisk (\*), the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

**c**

is the spool class assigned to the file.

**HOLD****NOHOLD**

is the user hold status associated with the file.

**KEEP****NOKEEP**

is the keep status associated with the file.

Response 10:

A tape drive specified on an SPXTAPE DUMP, LOAD, or SCAN command with a disposition of APPEND may be placed in a "WAITING" state:

- For DUMP, this occurs if the command with APPEND finishes processing before the next command is entered. The drive remains in this state until it is selected by another SPXTAPE DUMP command, or the operation is canceled or ended.
- For LOAD or SCAN, this occurs between the entry of the first command with APPEND and the command that completes the append sequence. The drive remains in this state until it is selected by a corresponding SPXTAPE LOAD or SCAN command with a disposition other than APPEND, or the operation is canceled or ended.

The following response is written in the command summary log and the volume log and displayed on the screen.

```
SPXTAPE {DUMP} WAITING ON VDEV vdev
        {LOAD}
        {SCAN}
```

**DUMP****LOAD****SCAN**

is the SPXTAPE operation.

***vdev***

is the virtual device number of the tape drive on which the SPXTAPE function is now waiting.

Action Taken on Tape Overflow or RUN (Rewind and Unload)

If either of the following conditions occurs, the tape drive enters the mount-pending state and response 5 is issued:

- On SPXTAPE DUMP, if the output tape fills before all the files selected on the command are written
- On SPXTAPE LOAD or SCAN with the RUN disposition, when SPXTAPE rewinds and unloads the tape after reading all the selected data.

You can either continue or end the SPXTAPE operation. See Response 5.

If all drives enter the mount-pending state and you take no action, the command remains dormant.

Action Taken on Error Conditions

The following sections summarize the action taken by SPXTAPE when various error conditions are encountered. Error messages are written in the command summary log.

*I/O Error during DUMP while Reading a File:* If an I/O error occurs during an SPXTAPE DUMP operation while reading a file from the spooling system:

1. The file is placed in system hold status.
2. The command continues with other selected files.

*I/O Error during DUMP while Writing a File:* If an I/O error occurs during an SPXTAPE DUMP operation while writing a file to tape:

1. The tape is rewound and unloaded.

2. The tape drive is placed in the mount-pending state.
3. The command continues by reprocessing all the files that were dumped to the volume that has the error. The files are dumped again, either to other active drives for this command (if any) or to this drive when a new tape is mounted.

*I/O Error during LOAD or SCAN while Reading a File:* If an I/O error occurs during an SPXTAPE LOAD or SCAN operation while reading a file from tape:

1. SPXTAPE tries to find the beginning of the next file on the tape by reading blocks until either a block is read without error or 10 consecutive bad blocks have been skipped.
2. If SPXTAPE finds a good block, the command continues. If SPXTAPE encounters 10 consecutive bad blocks, the tape is positioned according to the disposition specified on the command:
  - For LEAVE, the tape remains positioned after the last bad block. Response 4 is issued. The volume log indicates which files were successfully loaded or scanned. If this is the last or only active tape drive, the command completes with response 3. If the tape is a multiple dump tape (created by SPXTAPE DUMP with the LEAVE disposition), you can position it with the CMS TAPE command if you want to try loading or scanning the remaining files on the tape.
  - For REW, the tape rewinds but does not unload. Response 4 is issued. The volume log indicates which files were successfully loaded or scanned. If this is the last or only active tape drive, the command completes with response 3.
  - For RUN, the tape rewinds and unloads. Response 5 is issued. The volume log indicates which files were successfully loaded or scanned. You can mount another tape on this drive; if one or more other drives are active, you can do nothing; or you can enter the `spxtape end vdev` command to end processing on the device and complete the operation.

For recovery procedures for LOAD, see the section on recovering from an I/O error, cancel, or reset in the description of “[SPXTAPE LOAD](#)” on page 1846.

*I/O Error during LOAD while Writing a File:* If an I/O error occurs during an SPXTAPE LOAD operation while writing a file into the spooling system, SPXTAPE keeps trying to write the file until successful, or until spool space is exhausted.

*Spool Space Full:* If spool space becomes full during an SPXTAPE LOAD operation:

1. Partially loaded files are purged. The files that were successfully loaded are recorded in the volume log.
2. SPXTAPE LOAD terminates with responses 3 and 4.
3. The tape is positioned according to the disposition specified on the command.

Because of the spool space full condition, it is possible that the last part of the volume log and the command summary log may be lost.

If there is any doubt that there is insufficient spool space to do the SPXTAPE LOAD, it is highly recommended that you first enter the CP QUERY ALLOC SPOOL command and compare the results to the SPOOL PAGES field in response 3 in the command summary log created when the files were dumped. If that log is not available, use the SPXTAPE SCAN command to scan the volume, and use the information in response 3 from the scan.

*Spool File Limit Exceeded:* If the maximum spool file limit is reached during an SPXTAPE LOAD operation:

- If the owner of the file has reached the maximum number of spool IDs allowed for that virtual machine, SPXTAPE assigns the file to the user who entered the command, issues an error message, and continues processing.

If the issuer of the command has also reached the maximum spool ID limit, SPXTAPE continues processing the files without loading them, marking each file with NO\_LOAD in the SEG\_STAT field of the entry in the volume log.

- If the system maximum spool file limit has been reached, SPXTAPE issues an error message and terminates with responses 3 and 4. The tape is positioned according to the disposition specified on the command.

*System Failure:* If the system fails (stops) during an SPXTAPE DUMP, LOAD, or SCAN operation, the command summary log and volume logs are possibly incomplete. If the operation was SPXTAPE LOAD, files in the process of being loaded are purged by system restart.

Recovery of a DUMP or LOAD operation that was in progress can be accomplished as follows (after attaching new drives, if necessary):

- For DUMP with PURGE, discard the tape volumes not already rewound and unloaded prior to the failure, and reenter the SPXTAPE DUMP command with the same file selection criteria.
- For DUMP without PURGE, discard all the tape volumes already dumped and repeat the total process by reentering the SPXTAPE DUMP command.
- For LOAD, see the section on recovering from an I/O error, cancel, or reset in the description of the SPXTAPE LOAD command. Enter HELP SPXTAPE LOAD.
- For SCAN, discard the volume logs for the volumes actively being scanned at the time of the failure (the ones without response 4 at the end). Reenter the SPXTAPE SCAN command for these volumes.

## Messages

- HCP003E Invalid option - [command contains extra option(s) starting with] *option*
- HCP008E Invalid spoolid - *spoolid*
- HCP009E Invalid range - *range*
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP422E The same option was specified twice.
- HCP439E {USER *userid*|\*IMG|\*NLS|\*NSS|\*UCR} spool fileid limit exceeded
- HCP849E A userid must be specified with the spoolid
- HCP1001E An operand is missing for *option*.
- HCP1012E Queue type missing or invalid
- HCP1013E An invalid operand was supplied for *option* - *operand*.
- HCP1835E SPXTAPE command may only be issued on the base CPU
- HCP1836E SPXTAPE operation operand missing. Use DUMP, LOAD, SCAN, END, or CANCEL
- HCP1837E SPXTAPE device specification missing. Specify 'vdev' or 'vdev1-vdev2'
- HCP1838E Permanent I/O error on virtual device *vdev*. Attempting to read any remaining files
- HCP1839E File *userid spoolid2* (was *spoolid1* on tape) has been placed in system hold status because of a tape read I/O error
- HCP1840E File {*userid*|\*IMG|\*NLS|\*NSS|\*UCR} *spoolid* has been skipped because SPXTAPE could not read the control information. [The file has been placed in system hold status]
- HCP1841E Page read error while dumping *userid spoolid fn ft*. XAB data has been omitted on tape
- HCP1842E Page read error while dumping {*userid*|\*IMG|\*NLS|\*NSS|\*UCR} *spoolid fn ft*; *nnnnnnn* pages were not dumped to tape
- HCP1843E File {*userid*|\*IMG|\*NLS|\*NSS|\*UCR} *spoolid2* (was *spoolid1* on tape) has been {placed in system hold status|purged} because of pages omitted when the file was dumped
- HCP1844E Permanent error on virtual device *vdev*. Discard the volume and mount a new volume, or leave the drive not ready if other drives are active, or add another drive with the SPXTAPE DUMP command
- HCP1845E User {canceled|reset} last virtual device *vdev*; {DUMP|LOAD|SCAN} *vdev1*[-*vdev2*] has been canceled
- HCP1846E Maximum system spool file limit exceeded; LOAD *vdev1*[-*vdev2*] has been canceled
- HCP1847E Tape on virtual device *vdev* contains data that is not valid. The tape is positioned incorrectly or the volume is not in SPXTAPE format. Ending processing of this volume



- HCP1848E Spool space exhausted; LOAD *vdev1[-vdev2]* has been canceled
- HCP1849E SPXTAPE DUMP was not started. File selection criteria did not match any files
- HCP1850E User {canceled|reset} virtual device *vdev*. [Discard volume]
- HCP1900E SPXTAPE operation *operation* is not recognized. Use DUMP, LOAD, SCAN, END, or CANCEL
- HCP1901E *vdev\_field* is not a valid virtual device number or range; use 'vdev' or 'vdev1-vdev2'
- HCP1902E Virtual device *vdev1[-vdev2]* not attached
- HCP1903E Virtual device *vdev1[-vdev2]* not supported for SPXTAPE
- HCP1904E Virtual device *vdev1[-vdev2]* not active
- HCP1905E Virtual device *vdev1[-vdev2]* is not valid; *vdev3* is already active with SPXTAPE {DUMP|LOAD|SCAN}
- HCP1906E File {*userid*\*IMG|\*NLS|\*NSS|\*UCR} *spoolid2* (was *spoolid1* on tape) has been purged because {spool space is full|the tape was canceled}
- HCP1907E File {*userid*\*IMG|\*NLS|\*NSS|\*UCR} *spoolid2* (was *spoolid1* on tape) has been purged because of a DASD error. The file can be recovered by reloading it from tape
- HCP1908E File {*userid*\*IMG|\*NLS|\*NSS|\*UCR} *spoolid2* (was *spoolid1* on tape) has been {placed in system hold status|purged} because it was partially loaded. One or more tapes containing parts of the file were not loaded
- HCP1909E File dumped as *userid1 spoolid1* has been loaded to *userid2* as file *spoolid2*
- HCP1910E File {*userid*\*IMG|\*NLS|\*NSS|\*UCR} *spoolid2* (was *spoolid1* on tape) was partly loaded and then purged because of a tape read I/O error
- HCP1911W Tape on virtual device *vdev* not positioned at beginning-of-tape or after a tape mark. Attempting to process anyway
- HCP1912E Tape on virtual device *vdev* not positioned at beginning-of-tape or after a tape mark. Device *vdev* has been canceled
- HCP1913I SPXTAPE selection criteria includes system data files that will not be purged
- HCP1914E Virtual device *vdev1[-vdev2]* not write-enabled
- HCP1915E Operands beginning with *operand* conflict with an active SPXTAPE command
- HCP1916I File *userid queue spoolid* matches the SPXTAPE selection criteria but could not be dumped
- HCP1917E Tape device(s) {*rdev[-rdev]*|*vdev[-vdev]*} attached MULTIUSER.
- HCP6525E External Security Manager is unavailable.

## SPXTAPE CANCEL



### Authorization

Privilege Class: D, E, G

### Purpose

Use SPXTAPE CANCEL to cancel an SPXTAPE DUMP, LOAD, or SCAN operation on one or more tape drives.

### Operands

**vdev**

**vdev1-vdev2**

is the virtual device number of the tape drive, or the range of virtual device numbers, on which the SPXTAPE operation is canceled. If you specify a range, and all active tape drives in the range are operating for a single SPXTAPE command, they are canceled.

### Usage Notes

1. Use SPXTAPE CANCEL only when you want to *immediately* cancel the SPXTAPE operation on the specified tape drive or drives, even in the middle of a file. For example, use this command if a tape drive malfunctions in such a way as to be totally unavailable to you, or if you accidentally select a huge file that is going to take several tapes to dump. In contrast, SPXTAPE END allows processing to continue until the end of the current file.

2. When you receive response 8:

```
SPXTAPE CANCEL vdev initiated
```

SPXTAPE halts the I/O operation that was in progress. There are no further I/O operations to the canceled device by the SPXTAPE command currently processing.

3. No tape disposition is processed for a canceled tape drive. The tape drive and tape are left in the state that results from halting the I/O operation.
4. Only the user who entered the SPXTAPE DUMP, LOAD, or SCAN command can enter SPXTAPE CANCEL to cancel it.
5. If the specified range of virtual device numbers does not include all the tape drives associated with a single SPXTAPE command, only the selected drives are canceled. The original command continues to operate on the remaining drives.

If the range includes tape drives that are active for different SPXTAPE commands, none of the tapes are canceled. You must use separate SPXTAPE CANCEL commands. For example, if device 181 is active for one command and device 182 is active for another command, you cannot use a range of 181-182 to cancel them both at the same time.

6. If a canceled tape drive was being used by SPXTAPE DUMP, the files already dumped are discarded. Information about the files or pages processed on the canceled tape are not included in the command summaries. However, if there are other tape drives active for that SPXTAPE DUMP command, the files dumped to the canceled tape are dumped again to one of the other tapes.

7. If a canceled tape drive was being used by SPXTAPE LOAD, the files already loaded are not affected. If a file is in the process of being loaded when the tape drive is canceled, the file is purged from the system.
8. For additional usage information, see SPXTAPE [“Usage Notes”](#) on page 1820.

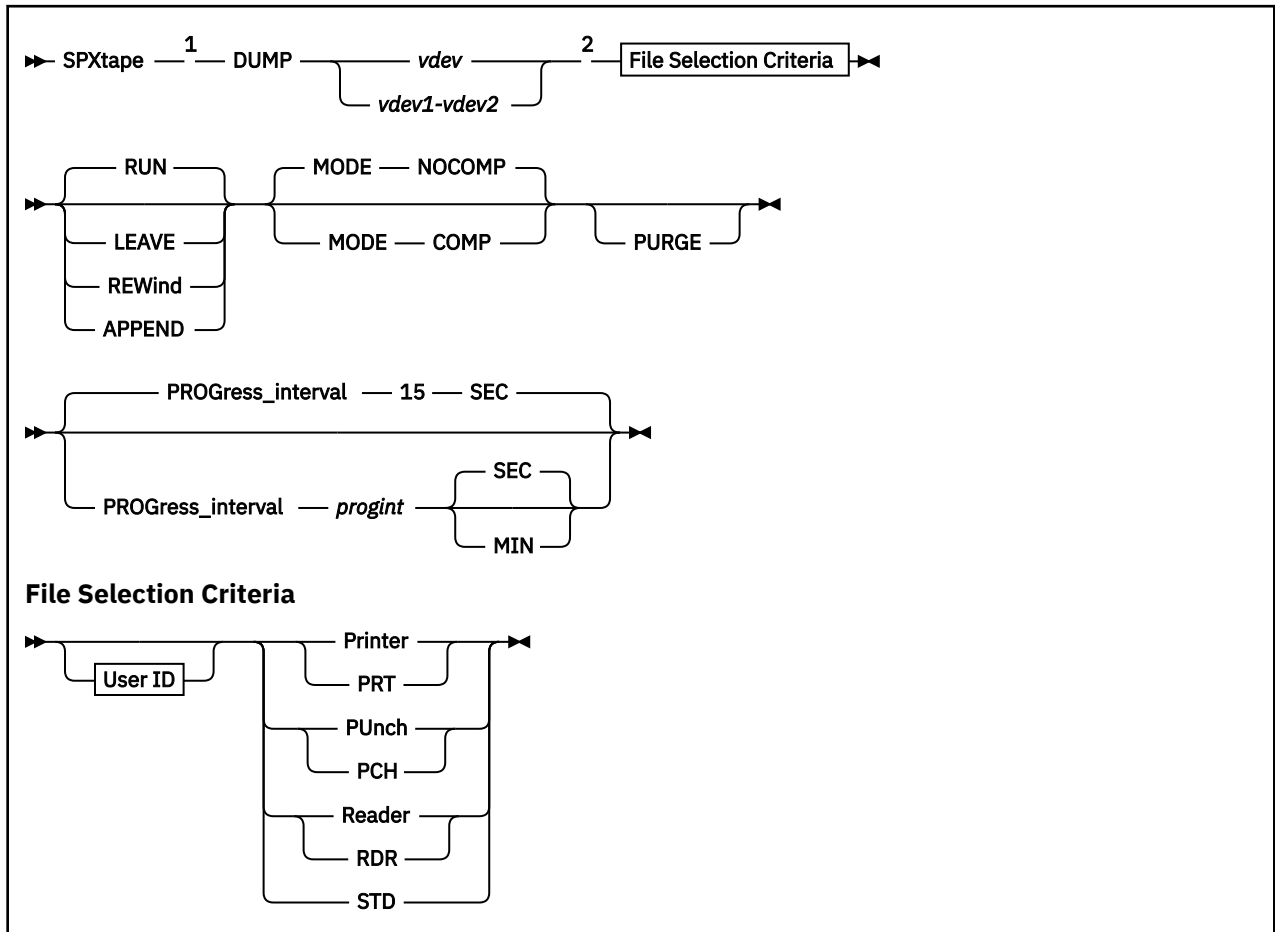
## Responses

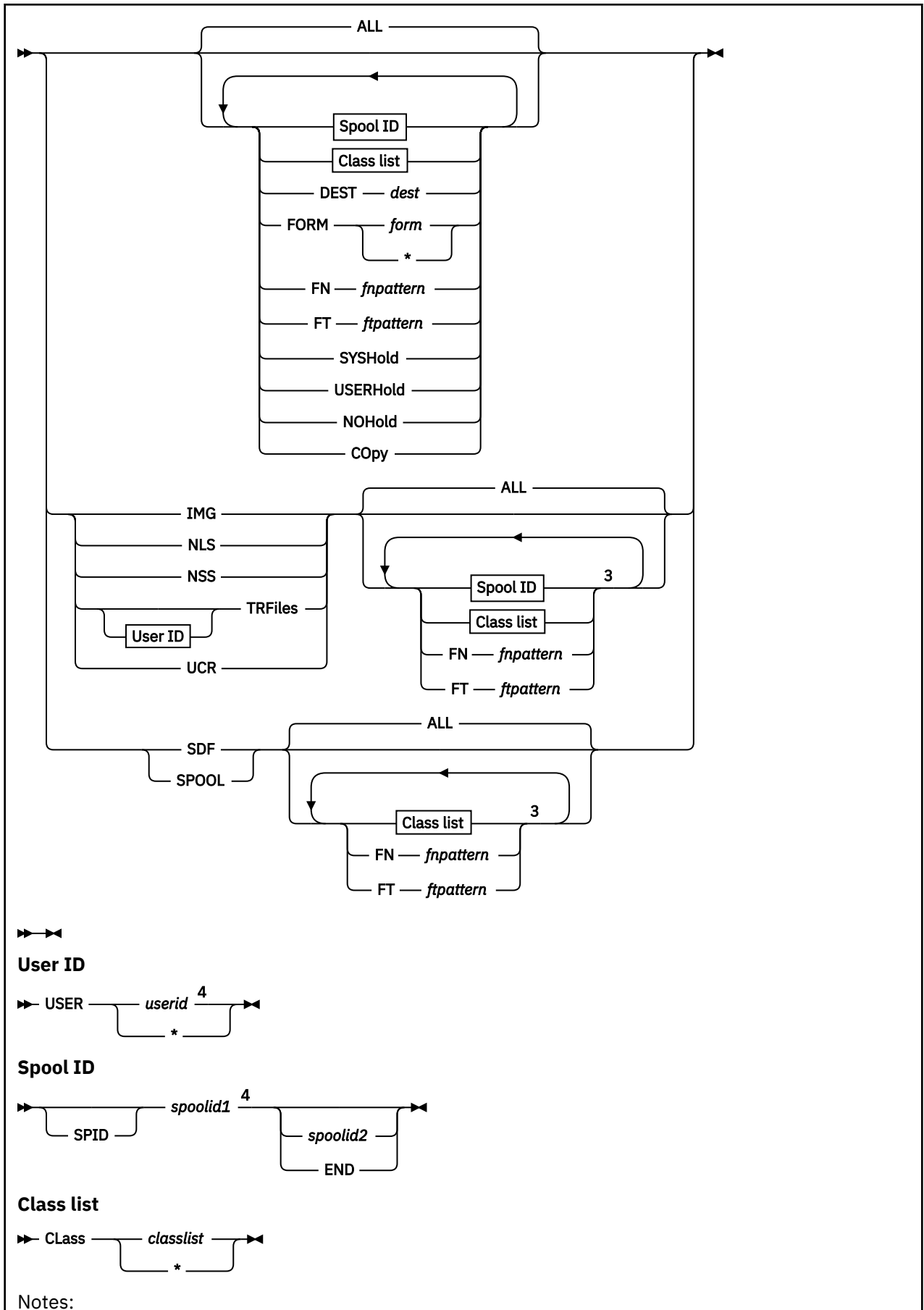
For response information, see SPXTAPE [“Responses”](#) on page 1821.

## Messages

For message information, see SPXTAPE [“Messages”](#) on page 1832.

# SPXTAPE DUMP





- <sup>1</sup> The status of an outstanding SPXTAPE operation can be determined using the QUERY VIRTUAL command. See the description of QUERY VIRTUAL TAPES for more details.
- <sup>2</sup> An abbreviated syntax is permitted for adding devices to a previous command.
- <sup>3</sup> You can specify file attribute selection operands in any order but you can specify each operand only once.
- <sup>4</sup> For a class D or E user, USER *userid* is required if you specify *spoolid1* with PRINTER, PRT, PUNCH, PCH, READER, RDR, STD, or TRFILES.

## Authorization

Privilege Class: D, E, G

## Purpose

Use SPXTAPE DUMP to read standard spool files and system data files from the spooling system and write them on tape.

## Operands

### **vdev**

#### **vdev1-vdev2**

is the virtual device number of the tape drive, or the range of virtual device numbers, where files are to be dumped. If you specify a range, any real supported tape drives attached as virtual device numbers in the range are used to contain the files being dumped.

Queue Selection Operands:

### **Printer**

#### **PRT**

selects printer spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are dumped.)

### **Punch**

#### **PCH**

selects punch spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are dumped.)

### **Reader**

#### **RDR**

selects reader spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are dumped.)

### **STD**

selects all types of standard spool files (printer, punch, and reader). Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are dumped.)

### **IMG**

selects image libraries. Class D or E users can use this operand.

### **NLS**

selects national language support files, such as message repository files. Class D or E users can use this operand.

### **NSS**

selects named saved systems and saved segments. Class D or E users can use this operand.

### **TRFiles**

selects system trace files. Class D, E, or G users can use this operand. (If you have only class G authority, only files owned by your user ID are dumped.)

### **UCR**

selects user class restructure files. Class D or E users can use this operand.

**SDF**

selects all types of system data files (IMG, NLS, NSS, TRFILES, and UCR). Class D or E users can use this operand.

**SPOOL**

selects all types of standard spool files and system data files. Because this includes all spooling system queues, only class D users can use this operand.

File Attribute Selection Operands:

**USER *userid*****USER \***

selects files owned by the specified user ID. An asterisk (\*) indicates your own user ID. (If you are a class G user, you cannot specify any entry except your own user ID or \*.)

If you are a class D or class E user, you must specify this operand if you are selecting standard spool files or system trace files by spool ID.

If you omit this operand, files are dumped regardless of user ID. (If you have only class G authority, only files owned by your user ID are dumped.)

**ALL**

selects files regardless of file attributes other than user ID. This is the default.

**SPID**

indicates the following value is a spool ID.

***spoolid1***

is the spool ID of a single file to be dumped or the start of a range of spool IDs. If you omit this operand, files are dumped regardless of spool ID.

If you are a class D or class E user, you must also specify the USER *userid* operand when selecting standard spool files or system trace files by spool ID.

***spoolid2***

is the end of the spool ID range.

**END**

indicates the spool ID range includes all spool IDs equal to or greater than *spoolid1*.

**Class *classlist*****Class \***

selects files with any of the specified classes. You can specify from one to eight classes (single alphanumeric characters with no intervening blanks).

For system data files, only certain classes are valid for each type of file. For information about system data file classes, see [z/VM: CP Planning and Administration](#).

If you specify an asterisk (\*) for the class list, or if you omit this entire operand, files are dumped regardless of class.

**DEST *dest***

selects files with the specified destination value, which may be from one to eight characters. If you omit this operand, files are dumped regardless of destination value.

**FORM *form*****FORM \***

selects files with the specified operator form name, which may be from one to eight characters. If you specify an asterisk (\*) for the name, or if you omit this entire operand, files are dumped regardless of form name.

**FN *fnpattern***

selects files that match the specified file name search pattern. The pattern may be a complete file name or a character string containing wild cards. For information about using wild cards, see usage note "6" on page 1841. You cannot use this operand to select *only* files that have *no* file name (whose file name field contains only blanks). If you omit this operand, files are dumped regardless of file name (including those with no file name).

**FT *ftpattern***

selects files that match the specified file type search pattern. The pattern may be a complete file type or a character string containing wild cards. For information about using wild cards, see usage note “6” on page 1841. You cannot use this operand to select *only* files that have *no* file type (whose file type field contains only blanks). If you omit this operand, files are dumped regardless of file type (including those with no file type).

**SYSHold**

selects system-held files. If you specify SYSHOLD but not USERHOLD or NOHOLD, only system-held files and files that are both system-held and user-held are dumped. If you do not specify SYSHOLD, USERHOLD, or NOHOLD, files are dumped regardless of hold status.

**USERHold**

selects user-held files. If you specify USERHOLD but not SYSHOLD or NOHOLD, only user-held files and files that are both user-held and system-held are dumped. If you do not specify USERHOLD, SYSHOLD, or NOHOLD, files are dumped regardless of hold status.

**NOHold**

selects files that are not held. If you specify NOHOLD but not SYSHOLD or USERHOLD, only files that are not held are dumped. If you do not specify NOHOLD, SYSHOLD, or USERHOLD, files are dumped regardless of hold status.

**COPY**

includes SSI cluster cross-system spool copy files in the dump. If COPY is not specified, only original files are dumped.

Tape Disposition Operands:

**RUN**

rewinds and unloads the tape after the dump completes. This is the default.

**LEAVE**

leaves the tape positioned to do another SPXTAPE operation.

**REWIND**

rewinds the tape after the dump completes but does not unload.

**APPEND**

appends the selection criteria specified on this command to the selection criteria specified on one or more *successive* SPXTAPE DUMP commands within one logical SPXTAPE DUMP operation. All commands in the append sequence must address the same tape devices and all except the last must be specified with the APPEND disposition. For additional information, see usage note “7” on page 1841.

Control Operands:

**MODE NOCOMP**

does not use the hardware compaction feature. This is the default. The density used is whatever the hardware defaults to.

**MODE COMP**

uses the hardware compaction feature. All tape drives within the virtual device number range specified on the command must have the hardware compaction feature. The density used is whatever the hardware defaults to.

**PURGE**

deletes files from the spooling system after they are successfully dumped. Only standard spool files are eligible for purging; system data files are not eligible. You can specify PURGE only on the SPXTAPE DUMP command that initiates a dump operation, not on any subsequent appends. For additional information, see usage note “9” on page 1842.



**PROGress\_interval 15 SEC****PROGress\_interval *progrint* SEC****PROGress\_interval *progrint* MIN**

allows the time interval between the progress reports to be changed from the default of 15 seconds. (See [Response 6](#).) The interval can be specified as a decimal number from 0 to 9999. Requesting an interval of 0 causes the progress reports to be suppressed.

If a logical SPXTAPE command is being created through the use of the APPEND option, the last explicitly requested interval will be used.

**Usage Notes**

1. Open and in-use files, with the exception of named saved systems and saved segments, cannot be dumped.
2. SPXTAPE does not dump class P named saved systems and saved segments or class W system trace files.
3. In an SSI cluster shared spool environment, printer and punch files locked on another system are not dumped.
4. A reader file being dumped is marked "in-use" and is therefore not available to the owner during the dump. Reader files are dumped even if they are locked on another system in an SSI cluster shared spool environment. However, if the owner of a reader file being dumped is logged on to another system in the SSI cluster (not where the file is being dumped), there is no indication to the owner that the file is not available.
5. Using SPXTAPE DUMP at off-peak hours on the system will have less impact on users due to I/O contention to the spool DASD. Also, with fewer users logged on, the availability of closed spool files for dumping is increased.
6. When using the FN or FT file attribute selection operand, you can use two special characters, \* (asterisk) and % (percent), as wild cards in the file name or file type pattern. These special characters have the following meaning:

**\***

is a placeholder character that represents any number of characters from zero to eight, depending on how many other characters are specified in the pattern. You can use as many asterisks as necessary anywhere in the pattern as long as the total number of characters, including the asterisks, does not exceed eight. For example, a pattern of *d\** means any string that begins with the character "D". However, a pattern of *\*d\** means any string that contains a "D", whether it is at the beginning, in the middle, or at the end. A pattern containing a single asterisk and no other characters means all strings regardless of length or character content (including the string that contains only blanks).

**%**

is a placeholder character that represents a single character. You can use as many percents as necessary anywhere in the pattern as long as the total number of characters, including the percents, does not exceed eight. For example, a pattern of *%%%* means any string that is three characters in length. A pattern of *%%%s* means any string that is four characters in length and ends with the character "S".

Wild cards can also be combined. For example, a pattern of *vm%%\** means any string four characters in length *or longer* that begins with the characters "VM".

7. Using the APPEND operand prevents the SPXTAPE DUMP operation from completing when all the files selected on the command have been processed. SPXTAPE waits for another DUMP command, addressing the same tape devices, either selecting more files or specifying a disposition other than APPEND. At that point, the files selected on this command might not have been written to tape yet, but might reside in a partially filled buffer.

SPXTAPE DUMP processing begun with the APPEND disposition completes only when you enter an SPXTAPE DUMP command, addressing the same tape devices, with a disposition operand of RUN, LEAVE, or REWIND (or no disposition, which results in a default of RUN).

A sequence of appended SPXTAPE DUMP commands are processed as one logical SPXTAPE DUMP command. This allows you (or an exec) to specify different sets of selection criteria for individual files or groups of files to be dumped on the same set of tapes in a single SPXTAPE DUMP processing run, as shown in the following sequence:

```
spxtape dump 300-399 selection_criteria_1 append
spxtape dump 300-399 selection_criteria_2 append
:
spxtape dump 300-399 selection_criteria_n run
```

It is not necessary to wait for the response to the first SPXTAPE DUMP command (response 4) before entering the next SPXTAPE DUMP command in the sequence.

8. In a sequence of appended SPXTAPE DUMP commands, if the file selection criteria specified on a new append overlap the file selection criteria specified on a previous append, duplicate files may be dumped. When looking for a file to dump, SPXTAPE does not check to see if the file has already been dumped.
9. When you use the PURGE operand, a file is purged only after all data has been successfully written to the tape volume (or volumes) that contain the file, and the file information has been recorded in the log. The log records for all the files on the volume are written after the trailer record is successfully written and the SPXTAPE disposition is performed. Therefore, the file is actually purged only after the volume that contains the end of the file is complete. Except in the case of a system failure, the log reflects what has been purged.  
  
In the system failure case (when the system stops while SPXTAPE DUMP is running), if the specified SPXTAPE disposition is RUN and any log data exists for the volume after warmstart, then all the data on the tape is good. However, all the corresponding purges may not have been done. You can use SPXTAPE SCAN to recover a log of what is actually on the tape. To determine what spool files you have to purge, you can use the logs obtained from scanning the dumped tapes (this is the most conservative method) or you can use the logs of the dump (this is a little less conservative).
10. If you specify PURGE on the first SPXTAPE DUMP command in an APPEND sequence, and you select system data files on that command or any subsequent commands in the sequence, the system data files are dumped but not purged.
11. If a dump is initiated to more than one tape drive and only a few files are being dumped, it is possible for an intervention-required condition on one of the drives to hold up the dumping of all the files. For example, if only two files are being dumped, the end of the first file and the beginning of the second file may fall in the same data block that SPXTAPE is writing to the first drive. If that drive goes into intervention-required status during the dumping of the first file, SPXTAPE waits for the intervention-required condition to be cleared. So the dumping of the second file cannot begin, even though another drive is available. No further dumping occurs until the first drive is taken out of the intervention-required state. No periodic progress report (response 6) is issued under these circumstances.
12. A logical SPXTAPE DUMP command, which may be a single command or a sequence of appended commands, dumps the selected files on the tape as a single tape file and writes a double tape mark at the end. SPXTAPE DUMP does not write tape marks between individual files.  
  
If an SPXTAPE DUMP command is entered with the LEAVE disposition, SPXTAPE backspaces over the second tape mark after dumping the files. Therefore, the result of using multiple SPXTAPE DUMP commands with the LEAVE disposition to dump files to a single tape is that the tape files are separated by single tape marks and the last tape file ends with a double tape mark. To load these tape files back into the spooling system, you must use multiple SPXTAPE LOAD commands.
13. For additional usage information, see [“Usage Notes” on page 1820](#).

#### Expanding the Device Range

You can add one or more tape drives to the device or range of devices specified on a previous SPXTAPE DUMP command that is still active. Enter another SPXTAPE DUMP command with an expanded range that includes all of the devices specified on the previous command plus the device or devices you want to add.

If the previous SPXTAPE DUMP command was entered with a disposition of LEAVE, REWIND, or RUN (or no disposition, which defaults to RUN), the command that you enter to expand the range is an abbreviated form that omits the selection criteria, disposition, and control operands:

```
spxtape dump vdev1-vdev2
```

If the previous SPXTAPE DUMP command was entered with the APPEND disposition, the command you enter to expand the range can include additional selection criteria. If you intend to append additional commands to the sequence, you must also specify APPEND on the command that expands the range. Otherwise, SPXTAPE interprets the command as the final one in the sequence.

For example, suppose the previous command was:

```
spxtape dump 300-349 std class a syshold append
```

If you want to add device 352 and also add class D system trace files to the selection criteria, enter:

```
spxtape dump 300-352 trfiles class d append
```

If you only want to add device 352, enter:

```
spxtape dump 300-352 append
```

Note that the range now also includes devices 350 and 351, if such devices are attached to you and are active. You cannot skip any active tape devices in the range. If you do not want to use devices 350 and 351, you must redefine or detach them before entering the SPXTAPE command.

## Responses

For response information, see SPXTAPE [“Responses”](#) on page 1821.

## Messages

For message information, see SPXTAPE [“Messages”](#) on page 1832.

## SPXTAPE END



### Authorization

Privilege Class: D, E, G

### Purpose

Use SPXTAPE END to end SPXTAPE DUMP, LOAD, or SCAN processing at the completion of the current file.

### Operands

**vdev**

**vdev1-vdev2**

is the virtual device number of the tape drive, or the range of virtual device numbers, on which the SPXTAPE operation is ended. If you specify a range, and all active tape drives in the range are operating for a single SPXTAPE command, they are ended.

### Usage Notes

1. Only the user who entered the SPXTAPE DUMP, LOAD, or SCAN command can enter SPXTAPE END to end the operation.
2. If the specified range of virtual device numbers does not include all the tape drives associated with a single SPXTAPE command, only the selected drives are ended. The original command continues to operate on the remaining drives.  
  
If the range includes tape drives that are active for different SPXTAPE commands, none of the tapes are ended. You must use separate SPXTAPE END commands. For example, if device 181 is active for one command and device 182 is active for another command, you cannot use a range of 181-182 to end them both at the same time.
3. You can use either SPXTAPE END or SPXTAPE CANCEL to remove failing drives or to make drives available for other work. END does it gracefully, at the completion of the current file; CANCEL does it immediately, even in the middle of a file.
4. The tape is positioned as specified by the disposition operand on the SPXTAPE command that initiated the activity being ended.

For LOAD or SCAN with the LEAVE disposition, the tape is positioned at an indeterminate record on the volume. To process another tape file on the same tape, enter TAPE FSF.

For DUMP with the APPEND disposition, if no other drives are active, the dump of the current file is completed and the DUMP command completes. This could leave many selected files not dumped. Therefore, you should not select all drives with SPXTAPE END unless the drives are in the "WAITING" status. If your intent is to have all the appended files complete, use SPXTAPE DUMP with REWIND, RUN, or LEAVE. This causes the command to complete when all the appended files have been dumped.

For LOAD or SCAN with the APPEND disposition, if the drive is in the "WAITING" status, no files are processed.

For LOAD, DUMP, or SCAN with the APPEND disposition, if the drive is active, SPXTAPE END changes the disposition to RUN; that is, the tape is rewound and unloaded.

Entering SPXTAPE END between tape reel mounts during a multiple reel load or scan of a single file is the way to bring an SPXTAPE LOAD or SCAN operation with the RUN disposition to completion.

5. For additional usage information, see [“Usage Notes” on page 1820](#).

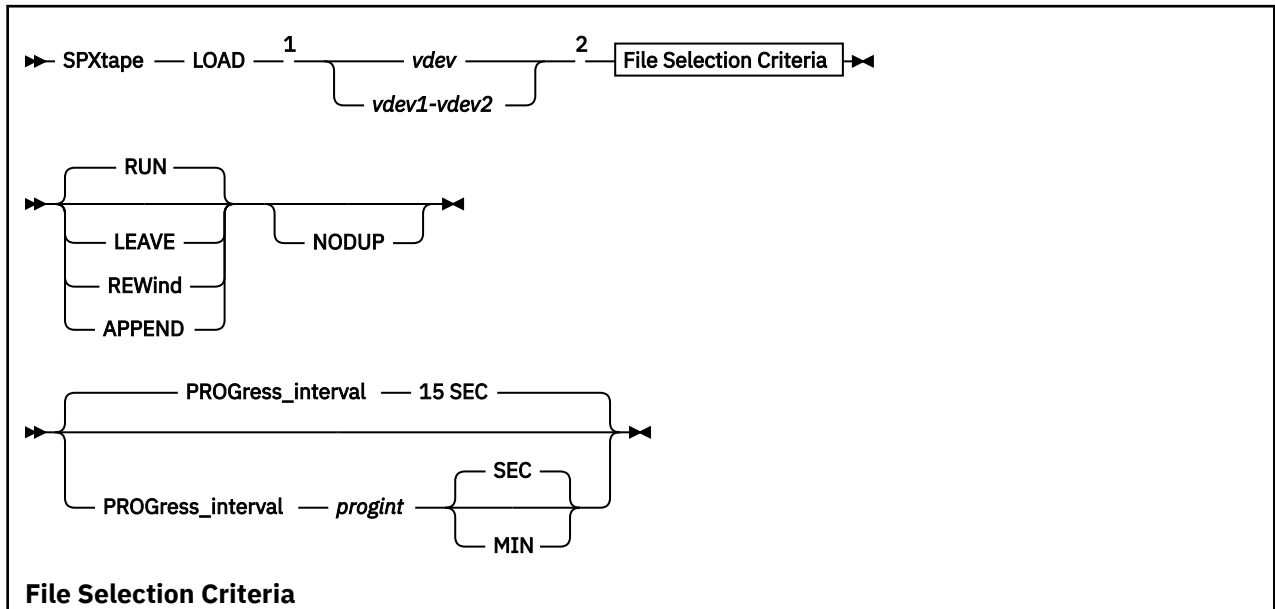
## **Responses**

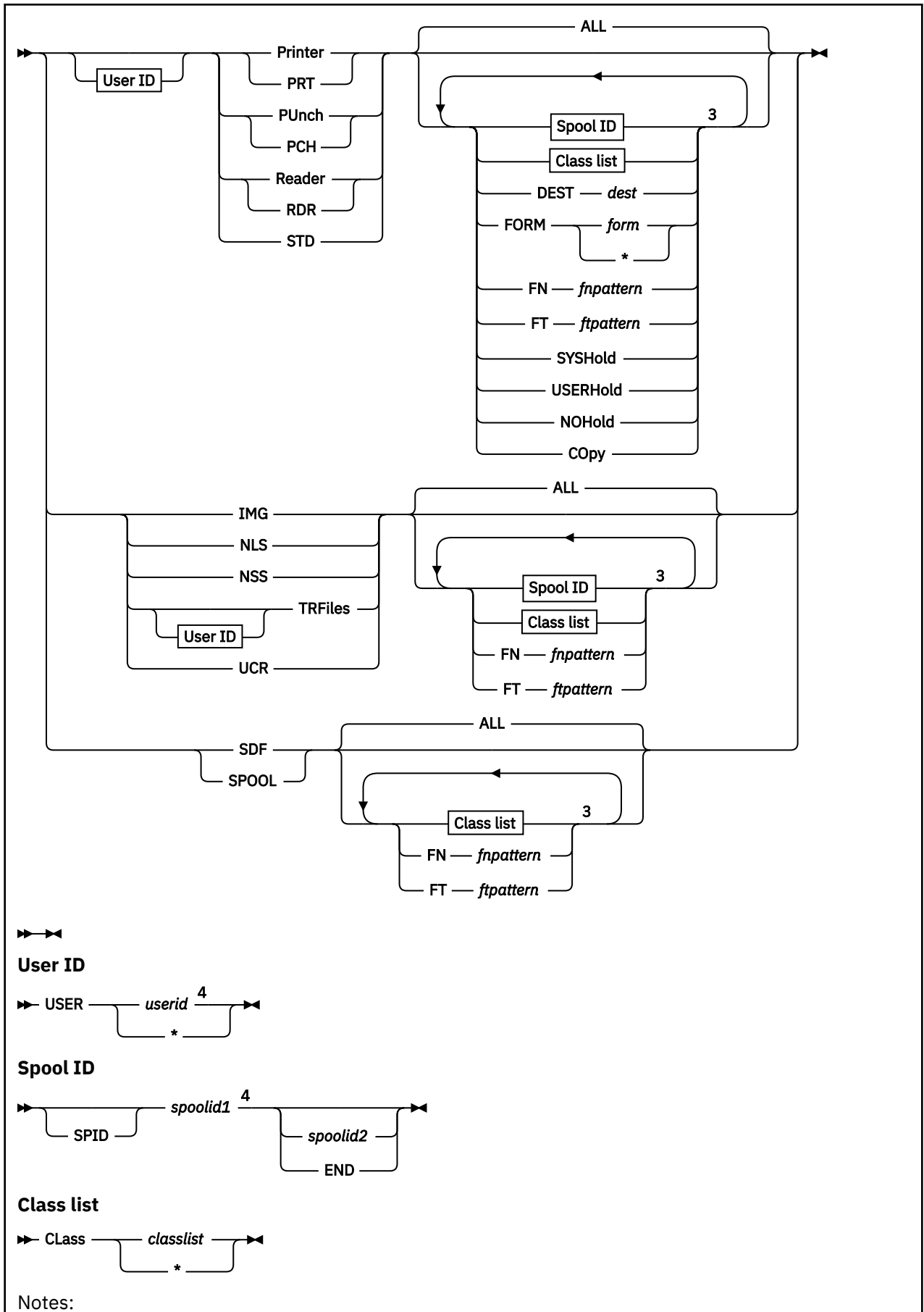
For response information, see SPXTAPE [“Responses” on page 1821](#).

## **Messages**

For message information, see SPXTAPE [“Messages” on page 1832](#).

## SPXTAPE LOAD





<sup>1</sup> The status of an outstanding SPXTAPE operation can be determined using the QUERY VIRTUAL command. See the description of QUERY VIRTUAL TAPES for more details.

<sup>2</sup> An abbreviated syntax is permitted for adding devices to a previous command.

<sup>3</sup> You can specify file attribute selection operands in any order but you can specify each operand only once.

<sup>4</sup> For a class D or E user, USER *userid* is required if you specify *spoolid1* with PRINTER, PRT, PUNCH, PCH, READER, RDR, STD, or TRFILES.

### Authorization

Privilege Class: D, E, G

### Purpose

Use SPXTAPE LOAD to read standard spool files and system data files from tape (that were written to tape by SPXTAPE DUMP) and write them into the spooling system.

### Operands

#### **vdev**

#### **vdev1-vdev2**

is the virtual device number of the tape drive, or the range of virtual device numbers, from which files are to be loaded. If you specify a range, any real supported tape drives attached as virtual device numbers in the range are used as the source for the files being loaded.

Queue Selection Operands:

#### **Printer**

#### **PRT**

selects printer spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are loaded.)

#### **Punch**

#### **PCH**

selects punch spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are loaded.)

#### **Reader**

#### **RDR**

selects reader spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are loaded.)

#### **STD**

selects all types of standard spool files (printer, punch, and reader). Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are loaded.)

#### **IMG**

selects image libraries. Class D or E users can use this operand.

#### **NLS**

selects national language support files, such as message repository files. Class D or E users can use this operand.

#### **NSS**

selects named saved systems and saved segments. Class D or E users can use this operand.

#### **TRFiles**

selects system trace files. Class D, E, or G users can use this operand. (If you have only class G authority, only files owned by your user ID are loaded.)

#### **UCR**

selects user class restructure files. Class D or E users can use this operand.



**SDF**

selects all types of system data files (IMG, NLS, NSS, TRFILES, and UCR). Class D or E users can use this operand.

**SPOOL**

selects all types of standard spool files and system data files. Because this includes all spooling system queues, only class D users can use this operand.

File Attribute Selection Operands:

**USER *userid*****USER \***

selects files owned by the specified user ID. An asterisk (\*) indicates your own user ID. (If you are a class G user, you cannot specify any entry except your own user ID or \*.)

If you are a class D or class E user, you must specify this operand if you are selecting standard spool files or system trace files by spool ID.

If you omit this operand, files are loaded regardless of user ID. (If you have only class G authority, only files owned by your user ID are loaded.)

The USER operand is checked for validity and will be rejected if the *userid* specified is not a user in the directory for the system that SPXTAPE LOAD is running on, and the *userid* does not have any SPOOL files on the system.

**ALL**

selects files regardless of file attributes other than user ID. This is the default.

**SPID**

indicates the following value is a spool ID.

***spoolid1***

is the spool ID of a single file to be loaded or the start of a range of spool IDs. If you omit this operand, files are loaded regardless of spool ID.

If you are a class D or class E user, you must also specify the USER *userid* operand when selecting standard spool files or system trace files by spool ID.

***spoolid2***

is the end of the spool ID range.

**END**

indicates the spool ID range includes all spool IDs equal to or greater than *spoolid1*.

**Class *classlist*****Class \***

selects files with any of the specified classes. You can specify from one to eight classes (single alphanumeric characters with no intervening blanks).

For system data files, only certain classes are valid for each type of file. For information about system data file classes, see [z/VM: CP Planning and Administration](#).

If you specify an asterisk (\*) for the class list, or if you omit this entire operand, files are loaded regardless of class.

**DEST *dest***

selects files with the specified destination value, which may be from one to eight characters. If you omit this operand, files are loaded regardless of destination value.

**FORM *form*****FORM \***

selects files with the specified operator form name, which may be from one to eight characters. If you specify an asterisk (\*) for the name, or if you omit this entire operand, files are loaded regardless of form name.

**FN *fnpattern***

selects files that match the specified file name search pattern. The pattern may be a complete file name or a character string containing wild cards. For information about using wild cards, see usage note [“2” on page 1851](#). You cannot use this operand to select *only* files that have *no* file name (whose file name field contains only blanks). If you omit this operand, files are loaded regardless of file name (including those with no file name).

**FT *ftpattern***

selects files that match the specified file type search pattern. The pattern may be a complete file type or a character string containing wild cards. For information about using wild cards, see usage note [“2” on page 1851](#). You cannot use this operand to select *only* files that have *no* file type (whose file type field contains only blanks). If you omit this operand, files are loaded regardless of file type (including those with no file type).

**SYSHold**

selects system-held files. If you specify SYSHOLD but not USERHOLD or NOHOLD, only system-held files and files that are both system-held and user-held are loaded. If you do not specify SYSHOLD, USERHOLD, or NOHOLD, files are loaded regardless of hold status.

**USERHold**

selects user-held files. If you specify USERHOLD but not SYSHOLD or NOHOLD, only user-held files and files that are both user-held and system-held are loaded. If you do not specify USERHOLD, SYSHOLD, or NOHOLD, files are loaded regardless of hold status.

**NOHold**

selects files that are not held. If you specify NOHOLD but not SYSHOLD or USERHOLD, only files that are not held are loaded. If you do not specify NOHOLD, SYSHOLD, or USERHOLD, files are loaded regardless of hold status.

Tape Disposition Operands:

**RUN**

rewinds and unloads the tape after the load completes. This is the default.

Upon detecting the logical end-of-tape (the end of the tape file written by SPXTAPE DUMP on that volume), SPXTAPE LOAD issues response 5, rewinds and unloads the tape, and waits for another volume to be mounted (the drive becomes "mount pending"). You must use the SPXTAPE END or SPXTAPE CANCEL command to cause SPXTAPE LOAD with the RUN disposition to complete.

**LEAVE**

leaves the tape positioned to do another SPXTAPE operation.

**REWind**

rewinds the tape after the load completes but does not unload.

**APPEND**

appends the selection criteria specified on this command to the criteria specified on one or more *successive* SPXTAPE LOAD commands within one logical SPXTAPE LOAD operation. All commands in the append sequence must address the same tape devices and all except the last must be specified with the APPEND disposition. For additional information, see usage note [“3” on page 1851](#).

Control Operands:

**NODUP**

prevents the loading of any file that would duplicate a file that already exists on the target spooling system queue. For additional information, see usage note [“4” on page 1851](#).

**PROGress\_interval 15 SEC****PROGress\_interval *progint* SEC****PROGress\_interval *progint* MIN**

allows the time interval between the progress reports to be changed from the default of 15 seconds. (See [Response 7](#).) The interval can be specified as a decimal number from 0 to 9999. Requesting an interval of 0 causes the progress reports to be suppressed.

If a logical SPXTAPE command is being created through the use of the APPEND option, the last explicitly requested interval will be used.

## Usage Notes

1. Using SPXTAPE LOAD at off-peak hours on the system will have less impact on users due to I/O contention to the spool DASD.
2. When using the FN or FT file attribute selection operand, you can use two special characters, \* (asterisk) and % (percent), as wild cards in the file name or file type pattern. These special characters have the following meaning:

**\***

is a placeholder character that represents any number of characters from zero to eight, depending on how many other characters are specified in the pattern. You can use as many asterisks as necessary anywhere in the pattern as long as the total number of characters, including the asterisks, does not exceed eight. For example, a pattern of d\* means any string that begins with the character "D". However, a pattern of \*d\* means any string that contains a "D", whether it is at the beginning, in the middle, or at the end. A pattern containing a single asterisk and no other characters means all strings regardless of length or character content (including the string that contains only blanks).

**%**

is a placeholder character that represents a single character. You can use as many percents as necessary anywhere in the pattern as long as the total number of characters, including the percents, does not exceed eight. For example, a pattern of %%% means any string that is three characters in length. A pattern of %%%s means any string that is four characters in length and ends with the character "S".

Wild cards can also be combined. For example, a pattern of vm%%\* means any string four characters in length *or longer* that begins with the characters "VM".

3. When using the APPEND operand, if there is no SPXTAPE LOAD command currently active on the addressed tape devices, the devices are placed in the append waiting state (response 10 is issued).

Actual loading from the tape devices does not commence until you enter an SPXTAPE LOAD command with a disposition of RUN, LEAVE, or REWIND (or no disposition, which results in a default of RUN).

A sequence of appended SPXTAPE LOAD commands are processed as one logical SPXTAPE LOAD command. This allows you (or an exec) to specify different sets of selection criteria for individual files or groups of files to be loaded from the same set of tapes in a single SPXTAPE LOAD processing run, as shown in the following sequence:

```
spxtape load 300-399 selection_criteria_1 append
spxtape load 300-399 selection_criteria_2 append
:
spxtape load 300-399 selection_criteria_n run
```

4. When using the NODUP operand, the definition of "duplicate" depends on the type of file.

For standard spool files and system trace files, "duplicate" means that all the file attributes of the file on the tape are identical to a file on the queue. This includes the time stamp for when the file was opened. An identical time stamp indicates that they are identical files. (If some files on the system have been imported from other systems, it is possible that files could have been created on both systems at exactly the same microsecond with all the same file attributes, but contain different data. However, this is not very probable.)

For system data files other than system trace files, "duplicate" means that the file name, file type, and class of the file on the tape are identical to the file on the queue. In addition, for named saved system and saved segment files, class A (active) and class R (restricted) files of the same file name and file type are considered duplicate because they cannot exist on the system at the same time.

A file that is found to be a duplicate is identified in the resulting volume log with DUP\_FILE in the SEG\_STAT field. The FILE field of this entry in the log contains the file ID of the file already on the

system. If the file on the system is moved to a different queue or user ID, or its file attributes are changed (for example, its spool class), the file is no longer identified as a duplicate.

The NODUP operand helps you recover if a system failure occurs while you are loading files from tape. If you add the NODUP operand to your selection criteria and reprocess all the tapes that have already been processed, all the files that loaded correctly before the system failure are skipped. However, even with NODUP specified, you may still receive duplicate files if you are loading the same file concurrently from more than one tape.

Keep in mind when using the NODUP operand that as the number of files in the queue where the files are being loaded increases, the amount of processing required to load the files from the tape increases dramatically. Each file to be loaded is compared to every file in the queue. For example, if the queue contains 1000 files and 1000 files are to be loaded, the result is over one million comparisons.

The consequence of loading duplicate files is more serious for system data files than for standard spool files. If the NODUP operand is not used when loading system data files (or all files) from a tape for which no dump log is available, you should check to make sure that duplicate files were not loaded. If this happens, check the files to determine which should be kept and purge the unnecessary files.

5. A logical SPXTAPE LOAD command, which may be a single command or a sequence of appended commands, loads one tape file. A tape file (which ends with one or more tape marks) contains one or more files dumped by one logical SPXTAPE DUMP command. If multiple tape files exist on the tape (written by multiple logical SPXTAPE DUMP commands), you must enter separate SPXTAPE LOAD commands to load each tape file.
6. Do not load the same tape more than once during processing of the same SPXTAPE command. Unpredictable results may occur, such as incomplete files, duplicate files, or loss of spool DASD space. However, there may be times when you need to reload the same volume. For example, if you encounter an I/O error while reading a tape, you might want to retry the operation. For more information, see the Recovering from an I/O Error, Cancel, Reset, or System Failure section in this command.
7. A file being loaded retains the same characteristics as the original file but is assigned a new spool ID in the spooling system. The new spool ID is obtained from the pool of IDs available to the recipient of the file.
8. If the file owner's user ID does not exist on the system to which the file is being loaded, or if the file owner runs out of spool IDs, the file is assigned to the user who issued the SPXTAPE LOAD command. If that user also runs out of spool IDs, the file is not loaded, and NO\_LOAD is recorded in the SEG\_STAT field of the entry in the volume log.
9. During SPXTAPE LOAD processing, response 5 is issued when all data has been read from a tape. Response 7, which is issued periodically to indicate the command is processing, may continue to be received even after response 5 has been issued for all the tapes being processed because the files are still being written into the spooling system. However, the command can no longer be ended or canceled at this point.
10. If you entered SPXTAPE LOAD with the RUN option (or no disposition, which results in a default of RUN), there is no way for SPXTAPE to know how many tape volumes are involved in the LOAD operation. Therefore, each time logical end-of-tape is reached, SPXTAPE issues Response 5 asking for the next volume to be mounted. Logical end-of-tape means the end of the tape file written by SPXTAPE DUMP on that volume.

If there are other tapes to be processed, mount the next tape. If you are using more than one tape drive, CP continues reading on the other drives. When you mount and ready the new tape, CP starts reading on that drive again.

If there are no other tapes to be mounted, and all tapes currently mounted have finished processing (you have received the mount next tape response for each drive), use the SPXTAPE END command to end the operation. A CANCEL command can be used in abnormal circumstances.

For more information, see the previous section on "Tape Disposition Operands" for SPXTAPE LOAD.

11. For additional usage information, see [“Usage Notes” on page 1820](#).
12. User Class Restructure (UCR) files that are loaded using SPXTAPE are no longer supported, and they will not be activated at the next IPL. Use the MODIFY statement or command to change command privilege classes. For information about MODIFY, see [Modifying a Command and Diagnose Code in z/VM: CP Exit Customization](#).

### Expanding the Device Range

You can add one or more tape drives to the device or range of devices specified on a previous SPXTAPE LOAD command that is still active. Enter another SPXTAPE LOAD command with an expanded range that includes all of the devices specified on the previous command plus the device or devices you want to add.

If the previous SPXTAPE LOAD command was entered with a disposition of LEAVE, REWIND, or RUN (or no disposition, which defaults to RUN), the command that you enter to expand the range is an abbreviated form that omits the selection criteria, disposition, and control operands:

```
spxtape load vdev1-vdev2
```

If the previous SPXTAPE LOAD command was entered with the APPEND disposition (in other words, you are in a sequence of SPXTAPE LOAD commands to be bundled into one logical command), the new command that you enter to expand the range can include additional selection criteria. If you do not also specify the APPEND disposition on the new command, SPXTAPE interprets the new command as the final command in the sequence.

For example, suppose the previous command was:

```
spxtape load 300-349 std class a syshold append
```

If you want to add device 352 and also add class D system trace files to the selection criteria, enter:

```
spxtape load 300-352 trfiles class d append
```

If you only want to add device 352, enter:

```
spxtape load 300-352 append
```

Note that the range now also includes devices 350 and 351, if such devices are attached to you and are active. You cannot skip any active tape devices in the range. If you do not want to use devices 350 and 351, you must redefine or detach them before entering the SPXTAPE command.

### Recovering from an I/O Error, Cancel, Reset, or System Failure

A file may be only partially loaded from a tape if:

- SPXTAPE LOAD encounters an I/O error while trying to read the file
- You cancel the SPXTAPE LOAD operation on the device where SPXTAPE is reading the file
- You reset the device (using RESET, DETACH, GIVE, or a similar CP command) where SPXTAPE LOAD is reading the file
- The system fails (stops) while SPXTAPE LOAD is reading the file.

Other files that follow on the tape volume may not be loaded at all.

Before attempting to recover any files from that volume, allow the load to continue on the other devices (if any) until the command completes, to load as many files as possible.

There are two ways to recover files from the volume:

- You can compare the volume log for the load to the volume log for the dump to see if files are missing or partially loaded. Then use the SPXTAPE LOAD command to load the files individually. If you are recovering from an I/O error, you must mount the tape on a different drive. Check all the volume logs for the dump to see if any file in question was dumped to more than one volume. You may also have to purge an incomplete version of the file that currently exists on the system in system hold status. This happens to a standard spool file if:

## SPXTAPE LOAD

- The file is partially loaded (the load is interrupted)
- The command ends on this volume before the file is read, but other parts of the file are loaded from other volumes.

If a system data file is incomplete, SPXTAPE purges it. When reloading a file, make sure you include all the volumes that contain parts of the file.

- You can reprocess the entire tape. If you are recovering from an I/O error, you must mount the tape on a different drive. Enter the SPXTAPE LOAD command for that drive, using your original selection criteria *plus* the NODUP operand.

If the reload completes without error, examine the volume log for the reload. Look for response 2 entries in which the SEG\_STAT field contains either PARTIAL or DUP\_FILE. If you find such an entry, check how the same file is recorded in the first volume log. (In some cases, there may be no entry in the first volume log.) All the possible combinations are shown in [Table 42 on page 1854](#), which also indicates if any additional actions are required. Additional actions are described following the table.

If you are using either of these methods to recover from an I/O error and the error is repeated, the tape has a bad spot. If you are reprocessing the entire tape, and SPXTAPE processing ends without loading any additional files after encountering the I/O error, either the bad spot extends to the end of the tape file or the spot is too large (10 consecutive bad records) for SPXTAPE to skip. Files in the bad records cannot be retrieved from the tape and must be recovered from some other source. However, all the files after the bad spot, either within the same tape file or in other tape files contained on the tape may be loaded. Physically position the tape past the damaged spot with the CMS command TAPE FSR or FSF, and enter the SPXTAPE LOAD command again. You may need to do this more than once to skip past the entire bad area. You may receive message HCP1911W as SPXTAPE skips to the start of where the next file was backed up, and resumes normal operation.

Type of File	First Load		Reload (NODUP)		Additional Actions Required?
	Load Results	Volume Log Entry	Load Results	Volume Log Entry	
Standard spool file, complete on this volume	Loaded	COMPLETE	Not loaded	DUP_FILE	No
	Partially loaded and placed in SYSHOLD	PARTIAL	If file on tape is <i>not</i> SYSHOLD, then loaded	COMPLETE	No, but you may want to purge the partially loaded file
			If file on tape is SYSHOLD, then not loaded	DUP_FILE	Yes
	Not loaded (not read)	No entry	Loaded	COMPLETE	No
System data file, complete on this volume	Loaded	COMPLETE	Not loaded	DUP_FILE	No
	Partially loaded and then purged	PARTIAL	Loaded	COMPLETE	No
	Not loaded (not read)	No entry	Loaded	COMPLETE	No

Type of File	First Load		Reload (NODUP)		Additional Actions Required?
	Load Results	Volume Log Entry	Load Results	Volume Log Entry	
Standard spool file, continued on one or more other volumes	Loaded	PARTIAL	Not loaded	DUP_FILE	Yes, if file on tape is SYSHOLD
	Partially loaded; when other parts of file on other volumes are loaded, incomplete file is placed in SYSHOLD	PARTIAL	If file on tape is <i>not</i> SYSHOLD, then loaded	PARTIAL	Yes
			If file on tape is SYSHOLD, then not loaded	DUP_FILE	Yes
	Not loaded (not read); when other parts of file on other volumes are loaded, incomplete file is placed in SYSHOLD	No entry	If file on tape is <i>not</i> SYSHOLD, then loaded	PARTIAL	Yes
			If file on tape is SYSHOLD, then not loaded	DUP_FILE	Yes
	System data file, continued on one or more other volumes	Loaded	PARTIAL	Not loaded	DUP_FILE
Partially loaded; when other parts of file on other volumes are loaded, incomplete file is purged		PARTIAL	Loaded and then purged	PARTIAL	Yes
Not loaded (not read); when other parts of file on other volumes are loaded, incomplete file is purged		No entry	Loaded and then purged	PARTIAL	Yes

*Additional Actions to Recover a File:* Follow these steps if [Table 42 on page 1854](#) indicates additional actions are required to recover a file.

1. Check the other volume logs (if any) for the same load to see if those volumes contain parts of the file.
2. If you find an entry in any log in which the file is recorded as COMPLETE, the file has been completely loaded. No further action is required.
3. If the file is recorded only as PARTIAL in other logs (or if this is the only volume log for the load), and the file is a standard spool file, purge the partially loaded file that exists on the system in system hold status. If the file is a system data file, SPXTAPE has already purged the partially loaded file.
4. Reload the file from all the volumes on which parts are located.

## SPXTAPE LOAD

*Recovering Multiple Files:* Instead of recovering each file separately, you can combine the loading of many individual files from various tape volumes into a single SPXTAPE LOAD operation, as long as all the files you want to load were dumped by one logical SPXTAPE DUMP command. Enter each SPXTAPE LOAD command (except the last) with the APPEND disposition operand. If you are using more than one tape drive, make sure you specify the complete device number range on each command. The load does not begin until you enter a command with a disposition other than APPEND. The order in which you mount the tapes does not matter.

### Responses

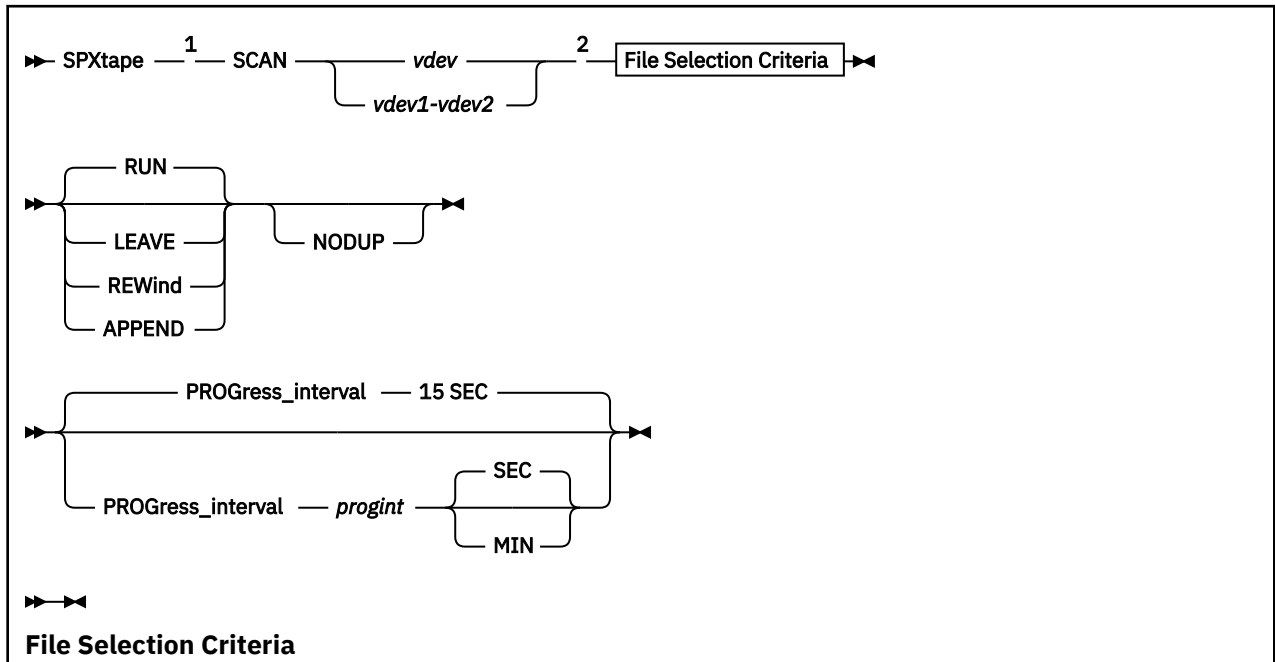
For response information, see SPXTAPE [“Responses”](#) on page 1821.

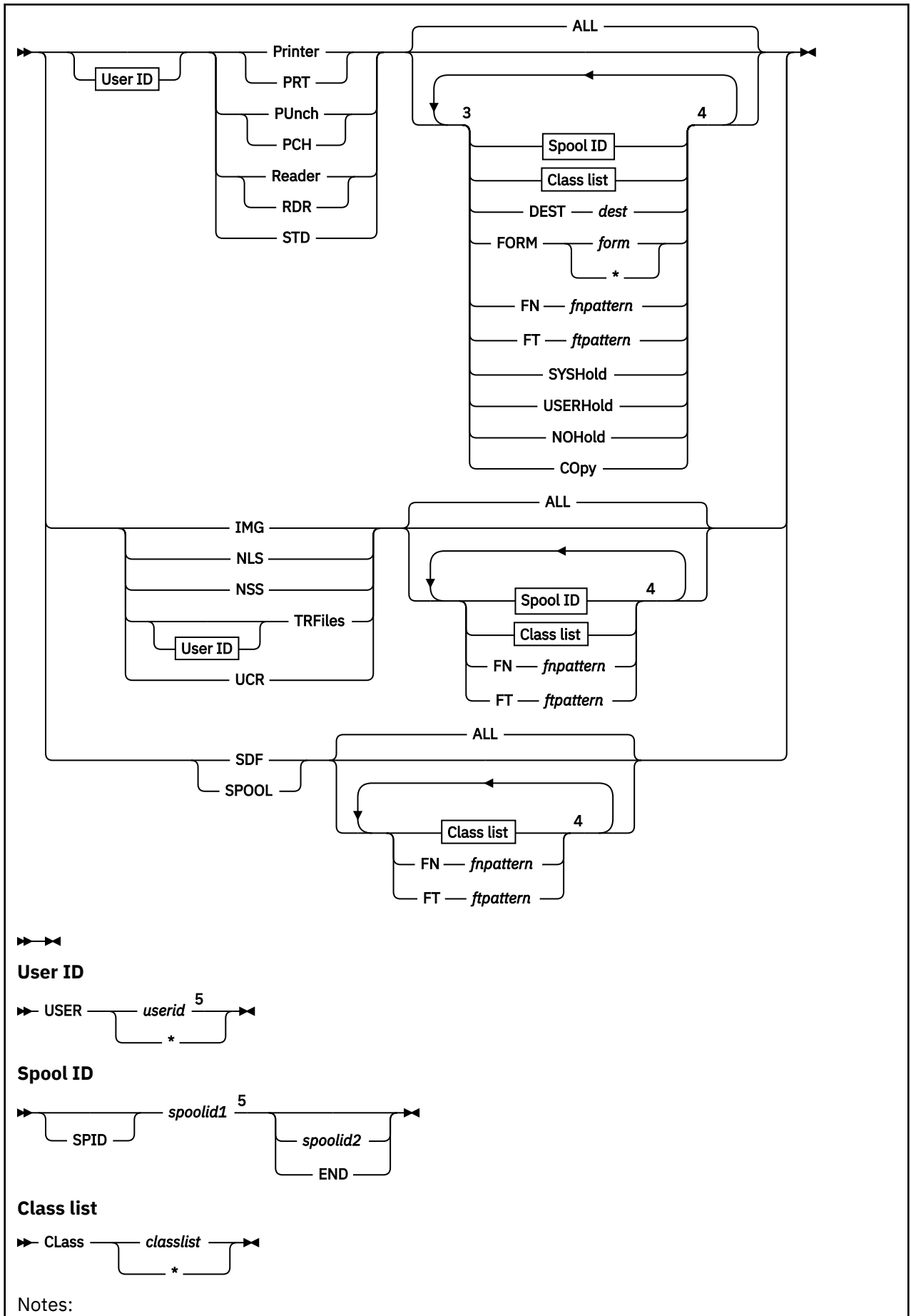
### Messages

For message information, see SPXTAPE [“Messages”](#) on page 1832.



# SPXTAPE SCAN





<sup>1</sup> The status of an outstanding SPXTAPE operation can be determined using the QUERY VIRTUAL command. See the description of QUERY VIRTUAL TAPES for more details.

<sup>2</sup> An abbreviated syntax is permitted for adding devices to a previous command.

<sup>3</sup> The status of an outstanding SPXTAPE operation can be determined using the QUERY VIRTUAL command. See the description of QUERY VIRTUAL TAPES for more details.

<sup>4</sup> You can specify file attribute selection operands in any order but you can specify each operand only once.

<sup>5</sup> For a class D or E user, USER *userid* is required if you specify *spoolid1* with PRINTER, PRT, PUNCH, PCH, READER, RDR, STD, or TRFILES.

## Authorization

Privilege Class: D, E, G

## Purpose

Use SPXTAPE SCAN to determine what standard spool files and system data files exist on tape (that were written to tape by SPXTAPE DUMP).

## Operands

**vdev**

**vdev1-vdev2**

is the virtual device number of the tape drive, or the range of virtual device numbers, where files are to be scanned. If you specify a range, any real supported tape drives attached as virtual device numbers in the range are used as the source for the files being scanned.

Queue Selection Operands:

**Printer**

**PRT**

selects printer spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are scanned.)

**PUnch**

**PCH**

selects punch spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are scanned.)

**Reader**

**RDR**

selects reader spool files. Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are scanned.)

**STD**

selects all types of standard spool files (printer, punch, and reader). Class D or G users can use this operand. (If you have only class G authority, only files owned by your user ID are scanned.)

**IMG**

selects image libraries. Class D or E users can use this operand.

**NLS**

selects national language support files, such as message repository files. Class D or E users can use this operand.

**NSS**

selects named saved systems and saved segments. Class D or E users can use this operand.

**TRFiles**

selects system trace files. Class D, E, or G users can use this operand. (If you have only class G authority, only files owned by your user ID are scanned.)

**UCR**

selects user class restructure files. Class D or E users can use this operand.

**SDF**

selects all types of system data files (IMG, NLS, NSS, TRFILES, and UCR). Class D or E users can use this operand.

**SPOOL**

selects all types of standard spool files and system data files. Because this includes all spooling system queues, only class D users can use this operand.

File Attribute Selection Operands:

**USER *userid*****USER \***

selects files owned by the specified user ID. An asterisk (\*) indicates your own user ID. (If you are a class G user, you cannot specify any entry except your own user ID or \*.)

If you are a class D or class E user, you must specify this operand if you are selecting standard spool files or system trace files by spool ID.

If you omit this operand, files are scanned regardless of user ID. (If you have only class G authority, only files owned by your user ID are scanned.)

**ALL**

selects files regardless of file attributes other than user ID. This is the default.

**SPID**

indicates the following value is a spool ID.

***spoolid1***

is the spool ID of a single file to be scanned or the start of a range of spool IDs. If you omit this operand, files are scanned regardless of spool ID.

If you are a class D or class E user, you must also specify the **USER *userid*** operand when selecting standard spool files or system trace files by spool ID.

***spoolid2***

is the end of the spool ID range.

**END**

indicates the spool ID range includes all spool IDs equal to or greater than *spoolid1*.

**Class *classlist*****Class \***

selects files with any of the specified classes. You can specify from one to eight classes (single alphanumeric characters with no intervening blanks).

For system data files, only certain classes are valid for each type of file. For information about system data file classes, see [Performing Data Management Services](#) in *z/VM: System Operation*.

If you specify an asterisk (\*) for the class list, or if you omit this entire operand, files are scanned regardless of class.

**DEST *dest***

selects files with the specified destination value, which may be from one to eight characters. If you omit this operand, files are scanned regardless of destination value.

**FORM *form*****FORM \***

selects files with the specified operator form name, which may be from one to eight characters. If you specify an asterisk (\*) for the name, or if you omit this entire operand, files are scanned regardless of form name.

**FN *fnpattern***

selects files that match the specified file name search pattern. The pattern may be a complete file name or a character string containing wild cards. For information about using wild cards, see Usage

Note “1” on page 1862. You cannot use this operand to select *only* files that have *no* file name (whose file name field contains only blanks). If you omit this operand, files are scanned regardless of file name (including those with no file name).

#### **FT *ftpattern***

selects files that match the specified file type search pattern. The pattern may be a complete file type or a character string containing wild cards. For information about using wild cards, see Usage Note “1” on page 1862. You cannot use this operand to select *only* files that have *no* file type (whose file type field contains only blanks). If you omit this operand, files are scanned regardless of file type (including those with no file type).

#### **SYSHold**

selects system-held files. If you specify SYSHOLD but not USERHOLD or NOHOLD, only system-held files are scanned (which includes any file that is both system-held and user-held). If you do not specify SYSHOLD, USERHOLD, or NOHOLD, files are scanned regardless of hold status.

#### **USERHold**

selects user-held files. If you specify USERHOLD but not SYSHOLD or NOHOLD, only user-held files are scanned (which includes any file that is both user-held and system-held). If you do not specify USERHOLD, SYSHOLD, or NOHOLD, files are scanned regardless of hold status.

#### **NOHold**

selects files that are not held. If you specify NOHOLD but not SYSHOLD or USERHOLD, only files that are not held are scanned. If you do not specify NOHOLD, SYSHOLD, or USERHOLD, files are scanned regardless of hold status.

Tape Disposition Operands:

#### **RUN**

rewinds and unloads the tape after the scan completes. This is the default.

Upon detecting the logical end-of-tape (the end of the tape file written by SPXTAPE DUMP on that volume), SPXTAPE SCAN issues response 5, rewinds and unloads the tape, and waits for another volume to be mounted (the drive becomes "mount pending"). You must use the SPXTAPE END or SPXTAPE CANCEL command to cause SPXTAPE SCAN with the RUN disposition to complete.

#### **LEAVE**

leaves the tape positioned to do another SPXTAPE operation.

#### **REWind**

rewinds the tape after the scan completes but does not unload.

#### **APPEND**

appends the selection criteria specified on this command to the criteria specified on one or more *successive* SPXTAPE SCAN commands within one logical SPXTAPE SCAN operation. All commands in the append sequence must address the same tape devices and all except the last must be specified with the APPEND disposition. For more information, see Usage Note “2” on page 1862.

Control Operands:

#### **NODUP**

prevents the scanning of any file that would duplicate a file that already exists on the spooling system queue where the file would be loaded. For more information, see Usage Note “3” on page 1862.

#### **PROGress\_interval 15 SEC**

#### **PROGress\_interval *progint* SEC**

#### **PROGress\_interval *progint* MIN**

allows the time interval between the progress reports to be changed from the default of 15 seconds. (See [Response 7](#).) The interval can be specified as a decimal number from 0 to 9999. Requesting an interval of 0 causes the progress reports to be suppressed.

If a logical SPXTAPE command is being created through the use of the APPEND option, the last explicitly requested interval will be used.

## Usage Notes

1. When using the FN or FT file attribute selection operand, you can use two special characters, \* (asterisk) and % (percent), as wild cards in the file name or file type pattern. These special characters have the following meaning:

**\***

is a placeholder character that represents any number of characters from zero to eight, depending on how many other characters are specified in the pattern. You can use as many asterisks as necessary anywhere in the pattern as long as the total number of characters, including the asterisks, does not exceed eight. For example, a pattern of d\* means any string that begins with the character "D". However, a pattern of \*d\* means any string that contains a "D", whether it is at the beginning, in the middle, or at the end. A pattern containing a single asterisk and no other characters means all strings regardless of length or character content (including the string that contains only blanks).

**%**

is a placeholder character that represents a single character. You can use as many percents as necessary anywhere in the pattern as long as the total number of characters, including the percents, does not exceed eight. For example, a pattern of %%% means any string that is three characters in length. A pattern of %%%s means any string that is four characters in length and ends with the character "S".

Wild cards can also be combined. For example, a pattern of vm%\* means any string four characters in length *or longer* that begins with the characters "VM".

2. When using the APPEND operand, if there is no SPXTAPE SCAN command currently active on the addressed tape devices, the devices are placed in the append waiting state (response 10 is issued).

Actual scanning on the tape devices does not commence until you enter an SPXTAPE SCAN command with a disposition of RUN, LEAVE, or REWIND (or no disposition, which results in a default of RUN).

A sequence of appended SPXTAPE SCAN commands are processed as one logical SPXTAPE SCAN command. This allows you (or an exec) to specify different sets of selection criteria for individual files or groups of files to be scanned on the same set of tapes in a single SPXTAPE SCAN processing run, as shown in the following sequence:

```
spxtape scan 300-399 selection_criteria_1 append
spxtape scan 300-399 selection_criteria_2 append
:
spxtape scan 300-399 selection_criteria_n run
```

3. When using the NODUP operand, the definition of "duplicate" depends on the type of file.

For standard spool files and system trace files, "duplicate" means that all the file attributes of the file on the tape are identical to a file on the queue. This includes the time stamp for when the file was opened. An identical time stamp indicates that they are identical files. (If some files on the system have been imported from other systems, it is possible that files could have been created on both systems at exactly the same microsecond with all the same file attributes, but contain different data. However, this is not very probable.)

For system data files other than system trace files, "duplicate" means that the file name, file type, and class of the file on the tape are identical to the file on the queue. In addition, for named saved system and saved segment files, class A (active) and class R (restricted) files of the same file name and file type are considered duplicate because they cannot exist on the system at the same time.

A file that is found to be a duplicate is identified in the resulting Volume Log with DUP\_FILE in the SEG\_STAT field. The FILE field of this entry in the log contains the file ID of the file already on the system. If the file on the system is moved to a different queue or user ID, or its file attributes are changed (for example, its spool class), the file is no longer identified as a duplicate.

Keep in mind when using the NODUP operand that as the number of files in the queue where the files would be loaded increases, the amount of processing required to scan the files on the tape increases dramatically. Each file being scanned is compared to every file in the queue. For example, if the queue contains 1000 files and 1000 files are scanned on the tape, the result is over one million comparisons.

4. A logical SPXTAPE SCAN command, which may be a single command or a sequence of appended commands, scans one tape file. A tape file (which ends with one or more tape marks) contains one or more files dumped by one logical SPXTAPE DUMP command. A logical SPXTAPE DUMP command may be a single command or a sequence of appended commands. If multiple tape files exist on the tape (written by multiple logical SPXTAPE DUMP commands), you must enter separate SPXTAPE SCAN commands to scan each tape file.
5. If you entered the SPXTAPE SCAN command with the RUN disposition (or no disposition, which results in a default of RUN), CP sends you a response when processing of a tape volume is complete that:
  - Asks you to mount the next tape on that device
  - Indicates the number of files and spool pages processed so far

If there are other tapes to be processed, mount the next tape. If you are using more than one tape drive, CP continues reading on the other drives. When you mount and ready the new tape, CP starts reading on that drive again. If there are no other tapes to be mounted, and all tapes currently mounted have finished processing (you have received the mount next tape response for each drive), use the SPXTAPE END command to end the operation.
6. For additional usage information, see [“Usage Notes” on page 1820](#).

### Expanding the Device Range

You can add one or more tape drives to the device or range of devices specified on a previous SPXTAPE SCAN command that is still active. Enter another SPXTAPE SCAN command with an expanded range that includes all of the devices specified on the previous command plus the device or devices you want to add.

If the previous SPXTAPE SCAN command was entered with a disposition of LEAVE, REWIND, or RUN (or no disposition, which defaults to RUN), the command that you enter to expand the range is an abbreviated form that omits the selection criteria, disposition, and control operands:

```
spxtape scan vdev1-vdev2
```

If the previous SPXTAPE SCAN command was entered with the APPEND disposition (in other words, you are in the middle of entering a sequence of SPXTAPE SCAN commands to be bundled into a single logical command), the command that you enter to expand the range can include additional selection criteria. However, you must also specify the APPEND disposition on your command. Otherwise, SPXTAPE interprets your command as the final command in the sequence.

For example, suppose the previous command was:

```
spxtape scan 300-349 std class a syshold append
```

If you want to add device 352 and also add class D system trace files to the selection criteria, enter:

```
spxtape scan 300-352 trfiles class d append
```

If you only want to add device 352, enter:

```
spxtape scan 300-352 append
```

Note that the range now also includes devices 350 and 351, if such devices are attached to you and are active. You cannot skip any active tape devices in the range. If you do not want to use devices 350 and 351, you must redefine or detach them before entering the SPXTAPE command.

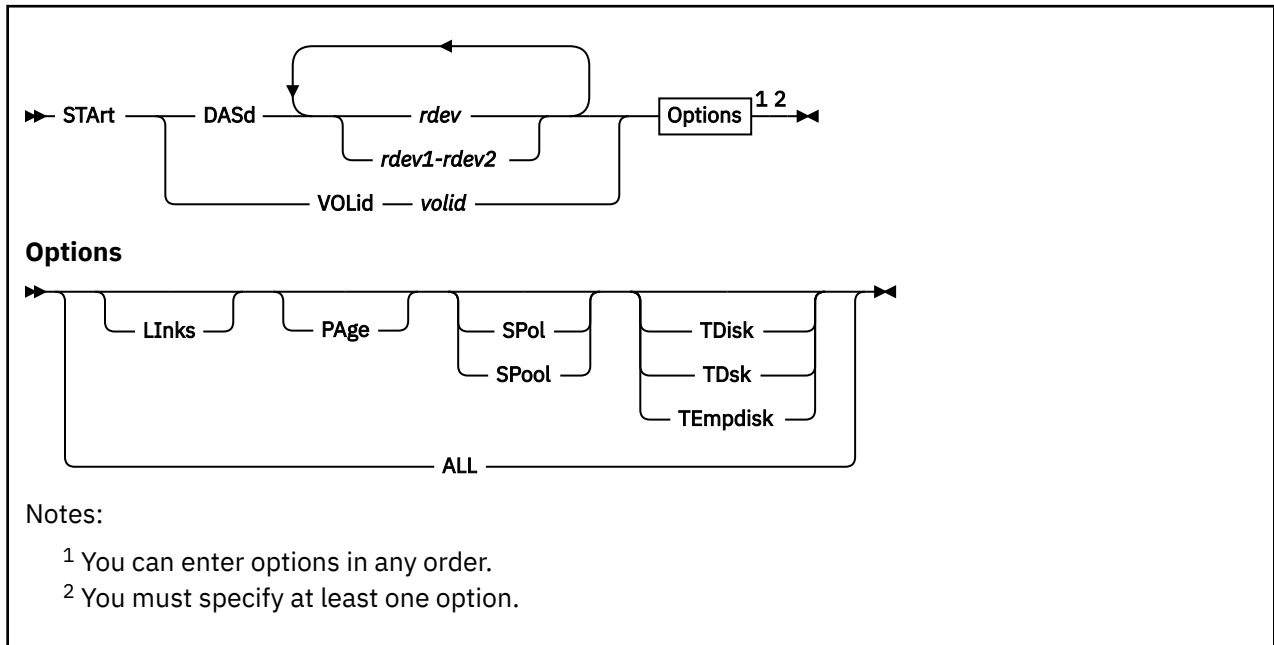
## Responses

For response information, see SPXTAPE [“Responses” on page 1821](#).

## Messages

For message information, see SPXTAPE [“Messages” on page 1832](#).

## START (Disk)



### Authorization

Privilege Class: B

### Purpose

Use START to restart (allow new allocations for) a device after it has been drained. You can also use it to change the processing options currently in effect for the device.

### Operands

#### DASd *rdev*

is the real device address of a particular device (or a range of DASD) you want to start. You can specify more than one real device.

#### VOLid *valid*

indicates you want to start a real DASD device by supplying its real volume serial number (*valid*).

### Options

#### ALL

means all of the following:

#### LInks

tells CP to allow users to link to minidisks on this device.

#### PAge

tells CP to write pages to this device during page-out operations.

#### SPol

#### SPool

tells CP to allocate space on this device for new spool records.



**TDisk****TDsk****TEmpdisk**

tells CP to allocate temporary minidisk space on this device.

**Responses**

Response 1:

```
Command Complete
```

- or -

```
nnnnnnnn device(s) specified; nnnnnnnn device(s)
successfully processed by START
```

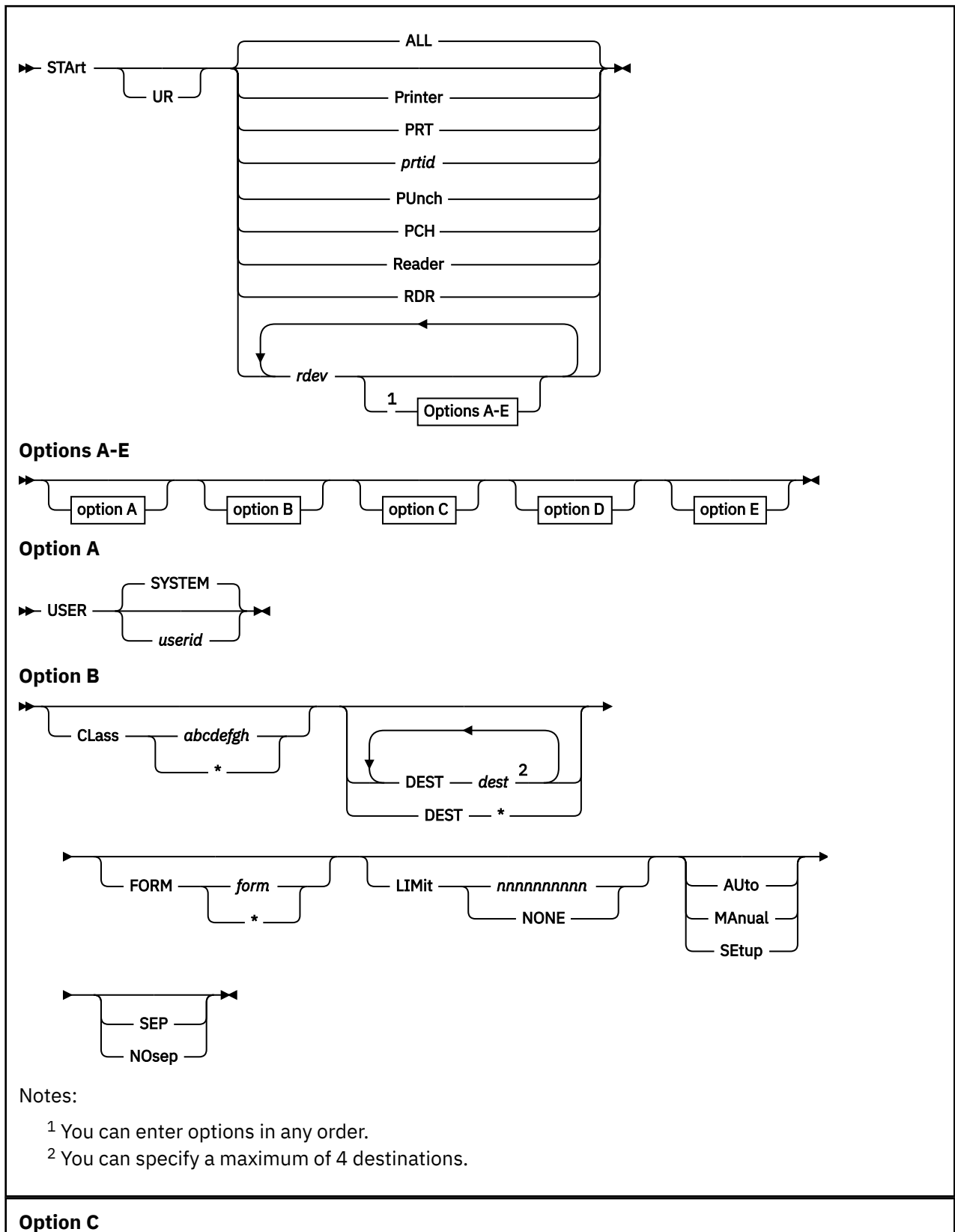
The following response may be sent to the system operator:

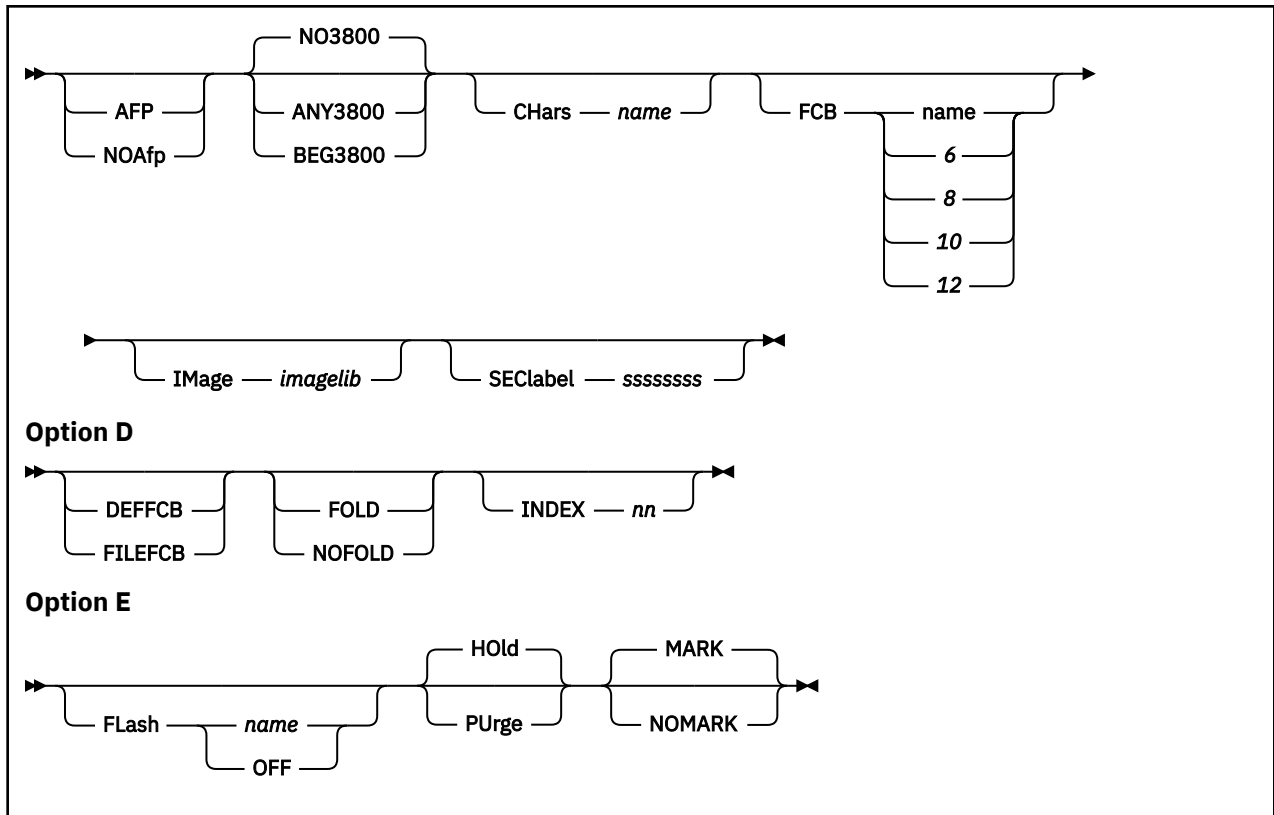
```
userid issued START command for volume volid on
device rdev for alloctype1 [alloctype2] [...]
```

**Messages**

- HCP002E Invalid operand - *operand*
- HCP021E A real device number was not specified or it is invalid.
- HCP023E VOLID missing or invalid
- HCP026E Operand missing or invalid
- HCP040E Device *rdev* does not exist
- HCP009E Invalid range - *range*
- HCP1004E *rdev* is not a DASD volume or is an invalid DASD type.
- HCP1127E {*rdev|volid*} is permanently drained.

## START UR (Unit Record)





## Authorization

Privilege Class: D

## Purpose

Use START to restart a drained spool device or to change the processing options in effect for the device.

## Operands

### UR

tells CP that you want to start a real unit record device.

### ALL

starts all of your readers, printers, and punches. ALL is the default.

### Printer

#### PRT

starts your printers.

#### *prt*id

tells CP to pass the entire command to the specified print server virtual machine (*prt*id) for processing. The *prt*id is provided by the print server virtual machine that issues the START command to an AFP. The variable *prt*id is 1 to 8 alphanumeric characters long. If you specify a *prt*id of 4 or fewer characters, *prt*id cannot contain all hexadecimal characters, because CP then interprets *prt*id as *rdev*.

### Punch

#### PUN

starts your punches.

#### *rdev* ...

is the real device number of the spooling device that you want to start. You can specify more than 1 real device, each with its own set of options.

### Reader

#### RDR

starts your readers.

### Options

Whenever a START command is issued with options, only the device settings that correspond to those options are changed. The other device settings remain unchanged from the previous START command. Device settings are saved across system warm starts. Cold, force, and clean start defaults are indicated under each option.

#### Option A

is valid for all real spooling devices.

#### Option B

is valid for printer and punch devices only, except for SETUP, which is valid only for impact printers.

#### Option C

is valid for both impact and 3800 printers.

**Note:** CHARS is not valid for 1403 printers without the UCS feature installed on 4245 or 4248 printers. FCB is not valid for a 1403 type printer.

#### Option D

is valid for impact printers only. DEFFCB and FILEFCB are valid for non-1403 impact printers.

#### Option E

is valid for 3800 printers only.

### USER *userid*

#### USER SYSTEM

specifies which virtual machines this device may service. If you specify *userid*, the spooling device prints or punches only that user's files. If the device is a printer or punch, the class designation must still match the class of the spool files that it services. If the device is a reader, no ID card is required.

If you specify SYSTEM, the spooling device is available to service spool files belonging to any user, which is the normal operation.

If no START command with the USER option has been issued since the last cold, force, or clean start, SYSTEM is the default.

### Class *abcdefgh*

#### Class \*

specifies the output classes the device can service. You can specify from 1 to 8 classes (each a single character, with no intervening blanks). If you are specifying multiple classes for your device, your files are serviced in the same order as the classes that you have specified. For example, if for your printer, you enter:

```
class ad1
```

all class A files are processed before class D files, and all class D files before class 1 files.

**Note:** The CLASS operand is not valid for a reader.

Specify a class of asterisk (\*) for printers or punches so that they can service any spool file class.

If no START command with the CLASS option has been issued since the last cold, force, or clean start, the default classes specified on the RDEVICE statement in the system configuration file.

### DEST *dest*

#### DEST \*

specifies the destinations that an output device can service. The operand *dest* is a 1- to 8-character value assigned to the printer. As many as 4 destination values can be specified.

Specify a DEST of asterisk (\*) to indicate that files are to be selected for processing regardless of the file's DEST setting.

If no START command with the DEST option has been issued since the last cold, force, or clean start, the default destinations specified on the RDEVICE statement in the system configuration file.

**FORM *form*****FORM \***

specifies the form that an output device can service. The operand *form* is a 1- to 8-character operator form number.

Specify asterisk (\*) for printers or punches so that files can be printed or punched regardless of form.

If no START command with the FORM option has been issued since the last cold, force, or clean start, the default form specified on the FORM\_DEFAULT statement in the system configuration file is used. If a default form was not specified, the default is STANDARD.

**LIMit *nnnnnnnnnn*****LIMit NONE**

specifies the number of records of a file to be printed or punched. The variable *nnnnnnnnnn* indicates the maximum number of records of a file to be printed or punched. You can specify up to 10 digits. NONE indicates there is no limit to the size of the file that can be printed or punched.

**AUto**

indicates that the operator wants CP to schedule forms output. The currently active form is processed first. When all files with the current form have been printed or punched, CP automatically selects a new form for the device. The operator receives a message to change forms, and the output device is idle. The operator can then put in the new form and press the START button (or enter the START UR command) to print files with the new form.

**MAnual**

indicates that the operator schedules output forms. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is printed or punched, the output device stops. The operator can then schedule a new form by issuing START with the FORM option.

If FORM (asterisk) \* is specified, spool files are printed or punched, regardless of the form, and the output device does not stop between spool files with different forms.

**SEtup**

indicates that the operator wants CP to print a page so that forms alignment can be verified. (This option is only valid for impact printers). The operator gets a message indicating that the form should be set up. When the form has been set up, the operator should press the STOP button, then the START button. An alignment page is then printed. The alignment page is the first nonblank page of the file, with letters replaced with Xs and numbers replaced by 9s. Each time the operator presses the STOP, START buttons after a full alignment page has been printed, another alignment page is printed. When the forms are aligned, issue the START command for the device. The printer returns to its previous mode (AUTO or MANUAL), and the file prints in its entirety.

If neither AUTO, MANUAL, nor SETUP is specified and the printer is currently in SETUP mode, it returns to the previous mode, AUTO or MANUAL. If no START command has been issued with the AUTO, MANUAL, or SETUP options since the last cold, force, or clean start, the default is MANUAL.

**SEP****NOsep**

indicates whether you want a file separator for output files. This option is not valid for readers.

If you *do not* want separator pages to be printed, you must always specify NOSEP.

If no START command with the SEP or NOSEP option has been issued since the last cold, force, or clean start, the default specified on the RDEVICE statement in the system configuration file.

**Note:** If an external security manager is installed and security label checking is enabled, the NOSEP option is not valid. For more information, see your security administrator.

### **AFP NOAfp**

specifies whether the printer is to process files with advanced function printer (AFP) characteristics. This option is valid for CP printers only. If a CP printer is started with the NOAFP option, that CP printer does not choose a file having an associated XAB or containing the special X'5A' CCWs. If a CP printer is started with the AFP option, that CP printer can select files having an associated XAB or containing the special X'5A' CCWs, but CP discards these attributes because they have no meaning for a CP printer.

If no START command with the AFP/NOAFP option has been issued since the last cold, force, or clean start, the default setting specified on the RDEVICE statement in the system configuration file.

### **ANY3800**

places no restrictions on the presence or absence of 3800 load CCWs within spool files. This option can potentially cause a slower output queue, depending on the number of 3800 load CCWs issued by the selected spool files.

### **BEG3800**

specifies that only spool files without 3800 load CCWs, or that contain 3800 load CCWs only at the beginning of the file, be printed on this device. This allows the user an initial setup with which the entire file is to be printed.

### **NO3800**

specifies that no spool files containing 3800 load CCWs be printed. Since load CCWs issued to 3800 printers can take a great deal of time, this option insures a fast output queue.

If no START command with the ANY3800, BEG3800, or NO3800 options has been issued since the last cold, force, or clean start, the default is NO3800.

### **CHars *name***

specifies the character set to be used by the printer when printing both the separator pages and the file. For 3800 printers, the character set specified is used for the separator pages and only those spool files that do not request a specific character set. The variable *name* is 1 to 4 characters long. This character set name must reside in the image library specified for the device.

For impact printers that support a character set, the CHARS specified is used for the entire file.

If no START command with the CHARS option has been issued since the last cold, force, or clean start, the default CHARS specified on the RDEVICE statement in the system configuration file.

**Note:** CHARS is not valid for 1403 printers without the UCS feature installed on 4245 or 4248 printers.

### **FCB *name***

#### **FCB 6, 8, 10, or 12**

specifies the forms control buffer used by the printer when printing both the separator pages and the file. For 3800 printers, the FCB specified is used for separator pages, and only those spool files that do not request a specific FCB.

For impact printers (except the 1403 which does not support the FCB), the FCB specified is used for the entire file.

The variable *name* is either a 1- to 4-character name of an FCB image in the image library specified for this device, or the number 6, 8, 10, or 12. This number tells the 3800 to print the entire spool file at 6, 8, 10, or 12 lines per inch, regardless of the size of the paper that is currently loaded.

**Note:** The number 10 is only supplied for 3800 Model 3 printers.

FCB values of 6, 8, 10, or 12 are only special cases on 3800 printers. If an FCB of 6, 8, 10, or 12 is specified for an impact printer, the START UR command tries to find an image library member with the corresponding name (for example, 32036 if FCB 6 is specified for a 3203 printer).

The number of lines to be printed on the separator page is determined by the FCB loaded on the printer. The default separator page contains 58 lines of data. If the page length defined by the FCB is less than the default separator page length, the separator page data must be customized in order

to fit on a single page. This may be done using CP Exit points that are provided in separator page processing. See *z/VM: CP Exit Customization* for details.

If no START command with an FCB option has been issued since the last cold, force, or clean start, the default FCB specified on the RDEVICE statement in the system configuration file.

### **IMage *imagelib***

specifies the name of the system image library file that contains the load images (character sets, FCBs, graphic character modifications, or copy modifications) to be used for printing files and separators. The variable *imagelib* is 1 to 8 characters long, and must be an existing image library file.

If no START command with the IMAGE option has been issued since the last cold, force, or clean start, the default image library file specified on the RDEVICE statement in the system configuration file.

### **SEClabel *sssssss***

specifies the security label of files to be printed on the printer. The variable *sssssss* is 1 to 8 alphanumeric characters that identify the SECLABEL of files that may be printed on the printer. The SECLABEL option is valid only when an external security manager (ESM) is installed and security label checking is enabled on your system.

### **DEFFCB**

uses the default FCB that was loaded in the printer. CP does not send imbedded LOAD FCB CCWs to the printer. The LOADBUF command specifies the default FCB. (DEFFCB is valid only for non-1403 impact printers.)

### **FILEFCB**

sends file FCBs to the real printer. File FCBs are LOAD FCB CCWs imbedded in the spool file. This is the default when you start the printer following a cold, force, or clean start. (FILEFCB is valid only for non-1403 impact printers.)

### **FOLD**

#### **NOFOLD**

specifies the loading of the universal character set (UCS) buffer with an operation code that permits printing of uppercase and lowercase bit configurations. If FOLD is specified, all characters are printed in uppercase. This is used when lowercase is specified in a file, but no lowercase exists in the UCS specified. If NOFOLD is specified, the characters are not translated into uppercase.

**Note:** If you specify the FOLD option, the control unit ignores bit positions 0 and 1 of the EBCDIC code when comparing the character codes from the UCS buffer and the print line buffer. Only uppercase characters are printed from either uppercase or lowercase data codes.



**Attention:** Misuse of the FOLD option can cause multiple characters to be printed for the same EBCDIC bit configuration.

### **INDEX *nn***

starts printing the output in print position *nn*. The variable *nn* is a decimal number from 0 to 31; a leading zero need not be specified. If you do not specify INDEX, the current setting of the indexing value is retained.

If you specify an INDEX of 0 then indexing is turned off and no indexing is performed. If you specify an indexing value from 1 to 31, then that value is used as the indexing value.

If you specify the INDEX keyword without a value, message HCP036E is issued. The LOADBUF command must be used to set the indexing value if you want to use the default indexing value coded in the FCB.

**Note:** INDEX is valid only for a device emulating a 3211 printer.

### **Flash *name***

#### **Flash OFF**

specifies the forms overlay negative loaded into the device. Spool files designated to be flashed with *name*, and spool files with no flash name associated with them, are selected for printing. The variable *name* is 1 to 4 alphanumeric characters long.

If OFF is specified, it indicates that no flash is to be used.

## START UR (Unit Record)

If no START command with the FLASH option has been issued since the last cold, force, or clean start, the default FLASH *name* specified on the RDEVICE statement in the system configuration file.

### **HOLD** **PURge**

specifies the system action to be taken on files with 3800 load errors. If HOLD is specified, all files with load errors are placed in system hold status. If PURGE is specified, all files with load errors are purged.

If no START command with the HOLD or PURGE option has been issued since the last cold, force, or clean start, the default is HOLD.

### **MARK** **NOMARK**

specifies whether a 3800 printer will mark 3800 separator trailer pages with separator bars.

If no START command with the MARK or NOMARK option has been issued since the last cold, force, or clean start, the default MARK setting specified on the RDEVICE statement in the system configuration file.

## Usage Notes

1. If a device is busy when you issue the START command, the changes that you specify take effect after the current file is complete.
2. If you specify the NOSEP operand for a punch, separator cards are not punched. Instead, for each file the operator receives the message:

```
PUN rdev START FOR OUTPUT
```

The punch can be started for the next file by pressing the START/STOP key on the device.

3. If a real spooling printer that is not a 3800 is started with the BEG3800 or ANY3800 options, all 3800 load CCWs contained in selected spool files are removed, and the file is printed with the available setup.
4. IBM does not recommend that you use spool file class to determine the output form (paper) for 3800 operation. If you change the paper on the 3800 printer without specifying a new form name with the FORM operand of the START command, 3800 errors result.  
  
Accordingly, when you change the paper on a 3800 printer, IBM recommends that you start the printer with the FORM operand and a new form name. If you prefer, you may instead drain the printer (with the DRAIN command) and enter the START command without the FORM operand.
5. If you are working within CMS and you want to enter a CP START command, use #CP START.
6. Throughout the START command operation, the 3800 printer specified refers to a CP-driven 3800 printer.
7. For printer IDs, the virtual machine associated with the printer ID processes the START command. Therefore, the options given on the START command must be valid for the virtual machine processing it.

When you enter either the START, START PRINTER, or START ALL commands, it causes all virtual machines associated with all print IDs to be started.

8. The keyword DEST is required to precede any destination value that is four or fewer hexadecimal characters or that matches the minimum truncation of an option on the START command.
9. If DEST is not specified on a command that sets a DEST value, the current value is unchanged.
10. A printer with DEST set to OFF only processes files with DEST specifically set to, or defaulted to, OFF.
11. If the START command is issued with the NOAFP option for a CP printer, the printer does not select files with advanced function printer (AFP) characteristics—that is, files that contain X'5A' CCWs or an extended attributes buffer (XAB). These files are not selected, even if other selection criteria are satisfied (for example, CLASS, FORM, DEST).



12. If you issue the START command for an output device that has been drained, or for an output device with a different FORM than the one previously started, you receive CP Message 0405. You should ensure that the correct form is mounted on the device; then, press the start button on the device or issue START from the console. Message HCP405 does not appear for an output device started with FORM asterisk (\*) or for one that was genned with FORM asterisk (\*) and started without using a different FORM.
13. If an error is detected loading an FCB or UCS to an impact printer, an error message is not issued until after the second START command. The first START command completes and the user is prompted to mount FORM.  
  
Also, if the printer is not ready, an error message is not issued until the printer is made ready.
14. CP prints the first nonblank page of the spool file as the alignment page when the printer is in SETUP mode. Pages of blanks that exist before the nonblank page will be processed. In this case, you will receive blank pages before the alignment page is printed.
15. When in SETUP mode, if all of the data in the spool file that is selected for printing is blanks, or if the first nonblank page contains only unprintable characters, then the alignment page results are unpredictable.
16. If the START command is issued with the FCB option, the LPP value will not change until the new FCB is actually loaded on the printer. The FCB is loaded, and the LPP value is changed, when the next file begins printing.

## Responses

Response for a Real Reader:

```
RDR  rdev STARTED {SYSTEM}
                    {userid}
```

Responses for a Real Printer:

When a printer is started, the first two lines of the response are:

```
PRT  rdev STARTED {SYSTEM} CLASS abcdefgh [SECLABEL ssssssss]
                    {userid}

PRT  rdev FORM operform {MANUAL} {SEP } { NO3800} [DEFFCB]
                        {AUTO } {NOSEP} {BEG3800} [FILEFCB]
                        {SETUP }      {ANY3800}
```

For 3800 printers, the third line is:

```
PRT  rdev FLASH name IMAGE imagelib CHARS name FCB {name} {HOLD } LPP nnn
                                                { 6 } {PURGE}
                                                { 8 }
                                                {10 }
                                                {12 }
```

For impact printers, the third line is:

```
PRT  rdev {NOFOLD} IMAGE imagelib CHARS chars [FCB name] [INDEX nn] LPP nnn
        {FOLD }
```

The fourth line of the response for all printers is:

```
PRT  rdev DEST {dest1} DEST {dest2 } DEST {dest3 } DEST {dest4 }
              {OFF }      {- NONE -} {- NONE -} {- NONE -}
```

The last line of the response for 3800 printers is:

```
PRT  rdev {AFP } {MARK }
          {NOAFP} {NOMARK}
```

The last line of the response for impact printers is:

```
PRT rdev {AFP }
        {NOAFP}
```

**rdev**

is the real device number of the device.

**SYSTEM****userid**

indicates which virtual machine's files the device can service.

**CLASS abcdefgh**

indicates which output classes the device can service.

**SECLABEL ssssssss**

indicates the SECLABEL of files to be printed on the printer. This is shown only when an ESM is installed and security label checking is enabled.

**FORM operform**

is the current form number processed on the device.

**MANUAL****AUTO****SETUP**

indicates the mode of forms processing for the printer or the punch.

**SEP****NOSEP**

indicates if file separators are to be produced between output files.

**NO3800****BEG3800****ANY3800**

specifies the restriction placed on 3800 load CCWs for spool files processed on a real printer. NO3800 means there are no load CCWs in the file. BEG3800 means only spool files without 3800 load CCWs, or with the CCWs at the beginning of the spool file are processed. ANY3800 means CP is to process any spool file, regardless of the occurrence of 3800 load CCWs.

**DEFFCB****FILEFCB**

specifies the FCB that is currently in use. DEFFCB means that the default FCB that was loaded in the printer is used. FILEFCB means that file FCBs are sent to the real printer.

**Note:** This only appears for non-1403 impact printers.

**FLASH name**

specifies the name of the forms overlay within the 3800 device.

**IMAGE imagelib**

indicates the image library containing load images (character sets, FCBs, graphic character modifications, or copy modifications) to be used for printing files and separators.

**CHARS name**

specifies the CHARS being used to print the file.

**FCB name****FCB 6****FCB 8****FCB 10****FCB 12**

indicates the FCB being used to print the file.

**HOLD****PURGE**

indicates the system action taken towards spool files that cause 3800 load errors.

**LPP *nnn***

indicates the lines per page that the printer is using for the generation of separator pages. The LPP value is determined from the FCB loaded on the printer.

If no FCB has been loaded to the printer, the LPP will appear as zero. Since the FCB is actually loaded when a file is printed, it is possible that the LPP will appear as zero or the LPP value for the prior FCB after the START command with the FCB operand is issued.

**Note:** If the number of lines per page derived from the FCB data is less than 30 or greater than 255, the LPP for that printer will be set to a default value of 66 for impact printers, or 60 for 3800 printers.

**NOFOLD****FOLD**

indicates if lowercase characters are being translated (folded) into uppercase for printing.

**IMAGE *imagelib***

indicates the image library being used by the printer.

**CHARS *chars***

indicates the character set being used by the printer.

**FCB *name***

indicates the FCB being used by the printer.

**INDEX *nn***

indicates the character position on the page at which to begin printing.

**DEST *dest***

is the destination value. The operand *dest* is a 1- to 8-character value. OFF is the default destination for the device for the first (of four) destination values. The indicator - NONE - is shown for destination values that have not been set.

**AFP****NOAFP**

specifies whether the printer is to process files with advanced function printer (AFP) characteristics.

**MARK****NOMARK**

indicates whether 3800 separator trailer pages are being marked with separator bars.

**Note:** CP itself does not provide a response for *prtids*. The virtual machine associated with the *prtid* provides the response.

Responses for a Real Punch:

```
PUN rdev STARTED {SYSTEM} CLASS abcdefgh
                   {userid}

PUN rdev FORM operform {MANUAL} {SEP }
                       {AUTO } {NOSEP}

PUN rdev DEST {dest1} DEST {dest2 } DEST {dest3 } DEST {dest4 }
           {OFF }      {- NONE -}      {- NONE -}      {- NONE -}
```

***rdev***

is the real device number of the device.

**SYSTEM*****userid***

indicates which virtual machine files the device may service.

**CLASS *abcdefgh***

indicates which output classes the device can service.

**FORM *operform***

is the current form number processed on the device.

**MANUAL****AUTO**

indicates the mode of forms processing for the punch.

**SEP  
NOSEP**

indicates whether file separators are produced between output files.

**DEST *dest***

is the destination value. The operand *dest* is a 1- to 8-character value. OFF is the default destination for the device for the first (of four) destination values. The indicator - NONE - is shown for other destination values which have not been set.

Responses for Currently Active Devices:

When the system begins processing a spool file that has the MSG attribute, the response is:

```
{RDR} rdev {READING } userid FILE spoolid
{PRT} rdev {PRINTING} userid FILE spoolid RECS {nnnn} COPY [*] nnn SEQ sss
{PUN}      {PUNCHING}      {nnnK}
                               {nnnM}
CL c typ
```

**RDR  
PRT  
PUN**

indicates the type of device on which the file is being processed.

***rdev***

is the real device number.

***userid***

is the name of the spool file owner.

**FILE *spoolid***

is the spool file *spoolid* number.

**RECS *nnnn***

**RECS *nnnK***

**RECS *nnnM***

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

**COPY [\*] *nnn***

specifies the number of file copies you want printed or punched when the file is produced.

An asterisk (\*) indicates that the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page. If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (\*) is ignored, and duplication occurs normally.

**SEQ *sss***

is the sequence number for the file on the printer or punch.

**CL *c***

is the spool file class.

***typ***

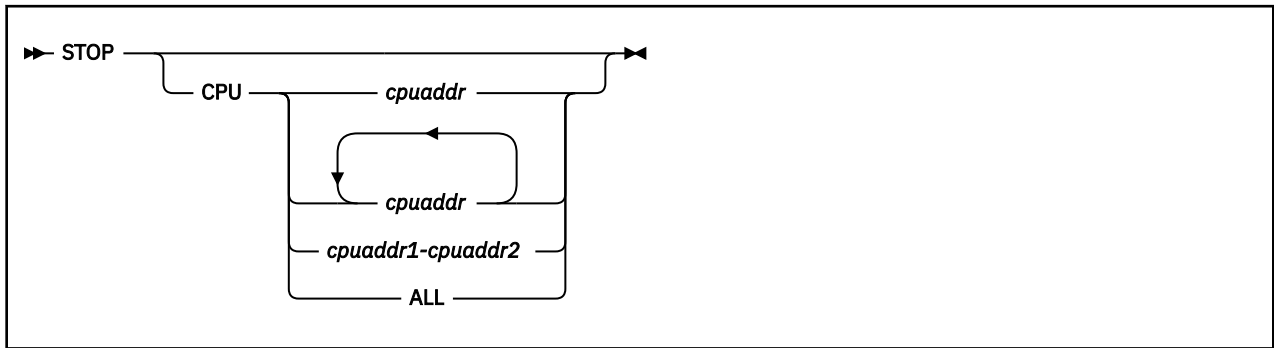
shows the device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP, or SYS.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP031E Buffer missing or invalid; [FCB too long|extended FCB not supported]

- HCP036E Index missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP140E *type {rdev|ldev}* attached to *userid*
- HCP148E Printer *rdev* buffer load *name* failed
- HCP241E An undefined load image *imagename* was specified for printer *rdev*.
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP418E {*rdev/vdev*} is not a spooled device.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*.
- HCP1552E More than {4|8} operands were specified for the {CHARS|CLASS} option.
- HCP1562E The length of buffer *bufname* is invalid.
- HCP2500E Image library *libname* could not be found [for printer *rdev*].
- HCP2501E A spool paging I/O error occurred while reading image [library] *imagename* on printer *rdev*.
- HCP6005E Option *option* is not supported by z/VM.
- HCP6525E Authorization request failed.

## STOP



### Authorization

Privilege Class: G

### Purpose

Use STOP to do the following:

- Soft-stop your virtual processors
- Hard-stop selected virtual processors.

### Operands

**CPU** *cpuaddr*

**CPU** *cpuaddr1-cpuaddr2*

**CPU** ALL

specifies the processor or processors you want to place in hard stop. You can specify a single processor address, a list of processor addresses delimited by blanks, a nonwrapping pair of addresses separated by a hyphen (-) with no blanks, or ALL. If you omit this operand, all of your virtual processors are placed into soft stop.

### Usage Notes

1. If you enter the STOP command without operands, all of your virtual processors are placed into soft stop.
2. This command is useful with a TRACE command to suspend execution when a certain condition is reached.
3. Execution of the virtual processor or processors is suspended until a BEGIN, SYSTEM RESTART, ATTN, IPL, EXTERNAL, or REQUEST command is entered.
4. The STOP command causes suspension of virtual machine execution, regardless of the SET RUN setting.

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP1454E CPU *nn* does not exist.
- HCP6026E The *command* command for CPU *nn* is invalid because the CPU is in check-stop state.

# STORE

---

## Purpose

Use STORE to alter any of the following locations in your real or virtual machine:

- First-level storage
- Second-level storage
- Third-level storage
- General registers
- Floating-point registers
- Control registers
- Storage keys
- The current program status word (PSW)

## Usage Notes

(General for Store)

1. Using a single command, you can store into the contents of multiple storage locations, registers, and control words. Separate each set of store requests from others by at least one blank. For example, if you want to store into hexadecimal location 40000 and the contents of register 5, enter **STORE 40000 10 G5 1F**.
2. If you enter a command line that contains an operand that is not valid, the STORE command terminates when it encounters that operand. Any previous valid operands are processed before termination occurs.
3. If you issue STORE with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the STORE is rejected and error message HCPSTT6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CLear option
  - SET MACHine (to a different machine architecture)

## Messages

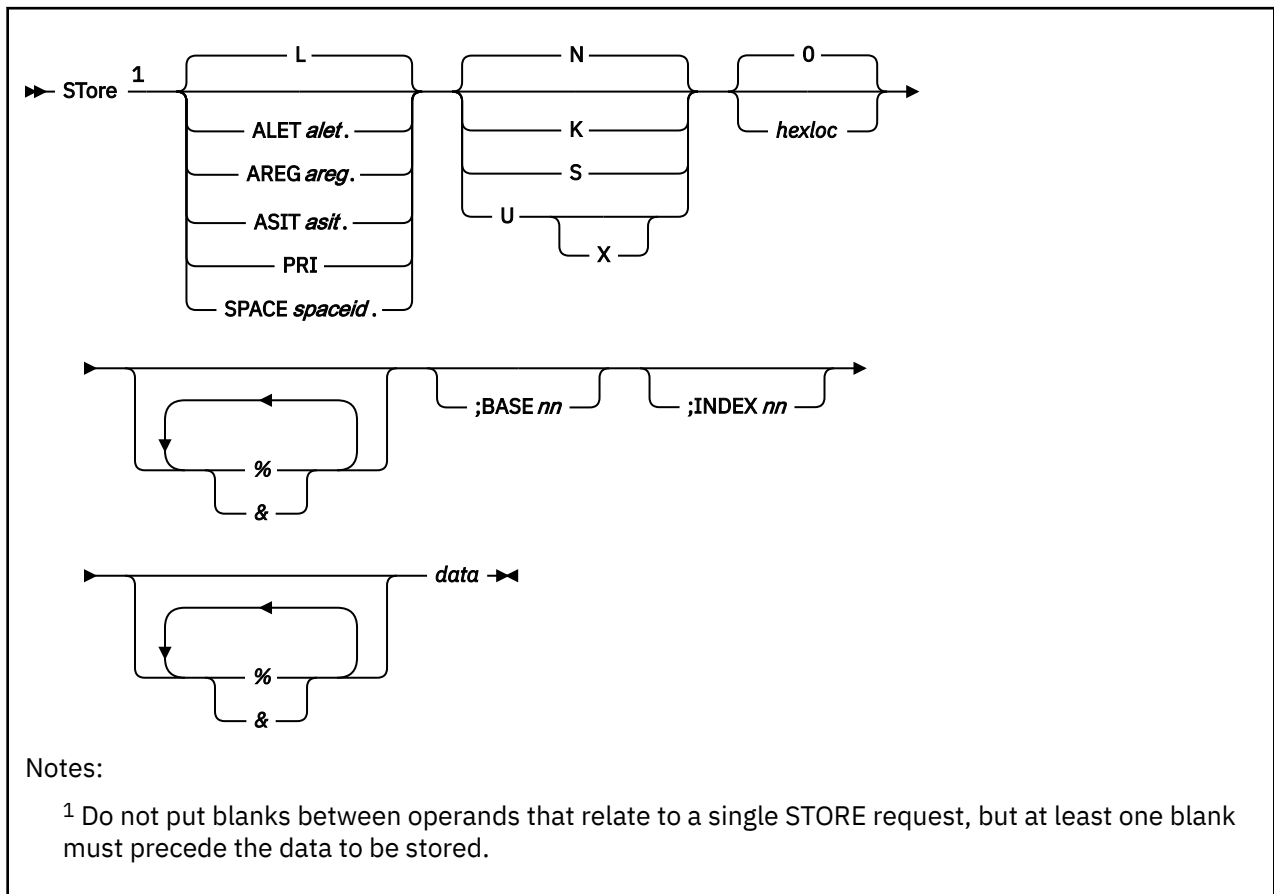
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP005E Invalid hexdata - *hexdata*
- HCP009E Invalid range - *range*
- HCP010E Invalid register - *register*
- HCP012E Invalid PSW - *psw*
- HCP026E Operand missing or invalid
- HCP033E Hexloc missing or invalid
- HCP060E {Dump failed|Printer output terminated}; virtual printer unavailable
- HCP160E Hexloc *hexloc* exceeds storage
- HCP163E STORE exceeds maximum register

## STORE

- HCP164E Hexloc *hexloc* non-addressable storage
- HCP260E {Command|*command* command} processing cannot complete due to errors.
- HCP261E *key* is an invalid storage key.
- HCP410E CP ENTERED; PAGING ERROR
- HCP422E The same option was specified twice.
- HCP657E The virtual machine's page zero is unavailable because it is damaged.
- HCP1001E An operand is missing for *option*.
- HCP1459E The virtual machine is placed in check-stop or system check-stop state due to a system malfunction with CPU *nn*.
- HCP2611E {Data|Key} cannot be stored into a read-only page.
- HCP6025E Operand {STATUS|STORE STATUS|RESTART} is invalid because CPU *nn* is in check-stop state.
- HCP6150E An option has been specified that is not valid in {ESA/390|ESA/XC|z/Architecture|z/XC} mode.
- HCP6151E Storage {operand} is not addressable because of { a protection | an addressing | a segment translation | a page translation | a translation specification | an ASN translation | an AFX translation | an ASX translation | an ALET translation | an ALEN translation | an ALE sequence | an ASTE validity | an ASTE sequence } exception.
- HCP6525E Authorization request failed.



## STORE (Guest Storage - ESA/XC, z/XC)



### Authorization

Privilege Class: G

### Purpose

Use STORE (Guest Storage - ESA/XC, z/XC) to store data into the storage of an ESA/XC or z/XC virtual machine.

### Operands

**L**

alters the contents of storage in an address space based on the setting of the address-space-control field in the guest PSW. This is the default.

If the virtual CPU is in primary-space mode, the contents of real storage in the host-primary address space are altered. If the virtual CPU is in the access-register mode and a nonzero BASE $nn$  operand is also specified, the contents of real storage in an AR-specified address space are altered.

If the virtual CPU is in access-register mode, but either the BASE $nn$  operand is not specified or is specified as 'BASE0', the contents of real storage in the host-primary address space are altered.

#### **ALET***alet.*

alters the contents of real storage in an access list entry token (ALET) specified address space. The *alet* may be 1 to 8 hexadecimal digits; leading zeros are optional. The ALET is used in conjunction with the virtual machine's host access list to determine the address space to store data in.

**AREG***areg*.

alters the contents of real storage in an access register specified address space. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The ALET within the access register is used in conjunction with the virtual machine's host access list to determine the address space to store data in.

**ASIT***asit*.

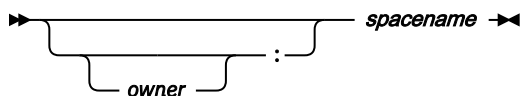
alters the contents of absolute storage in the address space identified by the specified address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

**PRI**

alters the contents of real storage in the host-primary address space. This is the address space from which the virtual CPU is fetching instructions at the time the command is run.

**SPACE***spaceid*.

alters the contents of absolute storage in the address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the address space, and *spacename* is the 1-character to 24-character address-space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

**N**

stores the data (in hexadecimal) in successive fullword locations starting at the address specified by *hexloc*. The smallest group of hexadecimal values that can be stored is one fullword. N is the default.

If *hexloc* is not on a fullword boundary, the store operation begins on the next lower fullword boundary. If *data* is less than one fullword (8 hexadecimal digits) in length, it is right-justified in the word and is filled in with zeros. If two or more hexadecimal words are specified, they must be separated with one or more blanks.

**K**

stores the data in the storage key for the 4 KB page in which *hexloc* is located. Two hexadecimal digits form the key for the page.

**S**

stores the data (in hexadecimal) without word alignment, starting at the address specified in *hexloc*. The shortest string that can be stored is 1 byte (2 hexadecimal digits). If the string contains an odd number of characters, the last character is not stored, an error message is issued, and the function terminates. If you specify the S operand, *data* must be a string of 2 or more hexadecimal digits with no embedded blanks.

**U**

stores the data starting at the address specified by *hexloc*. By default the data is treated as EBCDIC; if the X option is specified, the data is treated as ASCII. The data is stored in byte segments with no word alignment.

**hexloc**

specifies the starting storage location that is to be altered. The value *hexloc* can be 1 - 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The default is zero.

If neither the BASE or INDEX operand is used, *hexloc* is the address of the first location to be altered. This address is treated as a 64-bit address, regardless of the addressing mode of the virtual CPU.

If either the BASE or the INDEX operand is used, *hexloc* is a displacement rather than an address. The first location to be altered is specified by the sum of *hexloc* and the contents of the registers that are specified with the BASE and INDEX operands. If general register 0 is specified in the BASE or INDEX

operand, it is always treated as containing zero when forming this sum. The sum is treated as a 24-bit, 31-bit, or 64-bit address according to the addressing mode of the virtual CPU.

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is set to 0) are used as the address for the STORE command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the address for the STORE command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**BASEnn**

stores the contents of guest storage using the specified starting address (*hexloc*) as a displacement from the address in the specified general register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

If the virtual CPU is in access-register mode, and the L operand is specified (or defaulted), *nn* designates an access register/general register pair. Access register *nn* is implicitly used, in the same manner as if the AREG operand had been specified, in conjunction with the virtual machine's host access list to determine the address space in which to store. If the virtual CPU is not in the access-register mode, or if one of the other space-designation operands (PRI, AREG, ALET, SPACE, ASIT) is specified, access register *nn* is not implicitly used.

**INDEXnn**

stores the contents of guest storage using the specified starting address (*hexloc*) as a displacement from the address in the specified register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

**data**

is the data you want to insert into storage. The operand takes on different meanings depending on the type of STORE you are performing (N, S, K, or U). See the descriptions of each of these operands for the meaning of *data*.

## Usage Notes

- Special authorization is required to use this command to alter storage in any address space other than the host-primary address space of the virtual CPU. In order to use this command to alter storage in an address space, one of the following conditions must be satisfied:
  - Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.
  - If the address space is owned by another virtual machine, your virtual machine must have read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.

Authorization is granted on a space-by-space basis, so that you may be authorized to alter one address space of a virtual machine but may not be authorized to alter a different space.
- When running with a saved segment, you can store storage locations outside the range of your virtual machine size if they are within the saved segment. If locations exist between the upper limit of your virtual machine and the address at which the saved segment was saved, any attempt to store those locations (or associated keys) causes the non-addressable storage response to be displayed.

## STORE (Guest Storage - ESA/XC)

3. The operands L, ALET, AREG, and PRI use the host access-register-translation (ART) process. The ART process performs special-case translation processes for certain ALET values and access registers:
  - The ALET X'00000000' always indicates the host-primary address space.
  - Access register 0 is always treated as containing ALET X'00000000'. The contents of access register 0 are not used during host access-register-translation processing.
4. General register 0 is always treated as containing the value 0 when used as the base or index register through the BASE or INDEX operands. The contents of general register 0 are never used.
5. In the XC environment, special rules govern the conversion of real addresses into absolute addresses. In some cases, prefixing is performed as usual to convert a real address to an absolute address, while in other cases prefixing is not performed and real addresses are treated unchanged as absolute addresses.

These rules have an effect on the interpretation of the addresses that specify the range of storage locations to be altered:

- When storage is specified using the ALET or AREG operand and the ALET used is not X'00000000', prefixing is not performed on the addresses that specify the range of storage in the target address space.
  - When storage in the host-primary address space is specified by using the PRI operand, or by using the ALET or AREG operand when the ALET used is X'00000000', prefixing is performed as usual on the addresses that specify the range of storage in the host-primary address space.
6. Although the R operand is not shown in the syntax for this command, it is accepted for compatibility with the STORE (Guest Storage) command used for XA and ESA virtual machines. The R operand performs the same function as the L operand and alters storage in the host-primary address space.
  7. The BASE operand lets you store the storage used by the virtual machine in one less step than would otherwise be possible. Without the BASE operand, you may need to store the contents of a general register, then add the displacement, and then enter the DISPLAY command to examine storage the program is using. With the use of access registers, the BASE operand additionally lets you alter storage without needing to know whether the program is currently in the access-register mode. This is done automatically for you by CP.
  8. When multiple STORE requests are entered on a single line, the BASE and INDEX operands are not applied to subsequent STORE requests on the command line. They must be explicitly entered for each STORE request.
  9. During STORE command processing, key-controlled protection and low-address protection is not used. However, all of the host-related protection mechanisms, such as host page protection and those mechanisms applied during host-access-register translation, are used as normal. For example, an attempt to store at an address in a host-page-protected page, or to change the key at an address in a host-page-protected page, results in an error message.
  10. If the ALET or AREG operand is specified, the STORE command uses the host access-register-translation (ART) process to determine the address space in which to store data. This translation process may fail because of one of several program exception conditions. If the translation does not succeed, the current request is rejected and processing of the command line is ended. The rejected operands are displayed along with a message indicating the nature of the exception condition.
  11. When the ASIT or SPACE operand is specified, the STORE request fails if the specified address space does not exist, or if you are not authorized to access the address space. The current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
  12. If you use an underscore, it must be followed by 8 hexadecimal digits.

## Responses

Response 1:

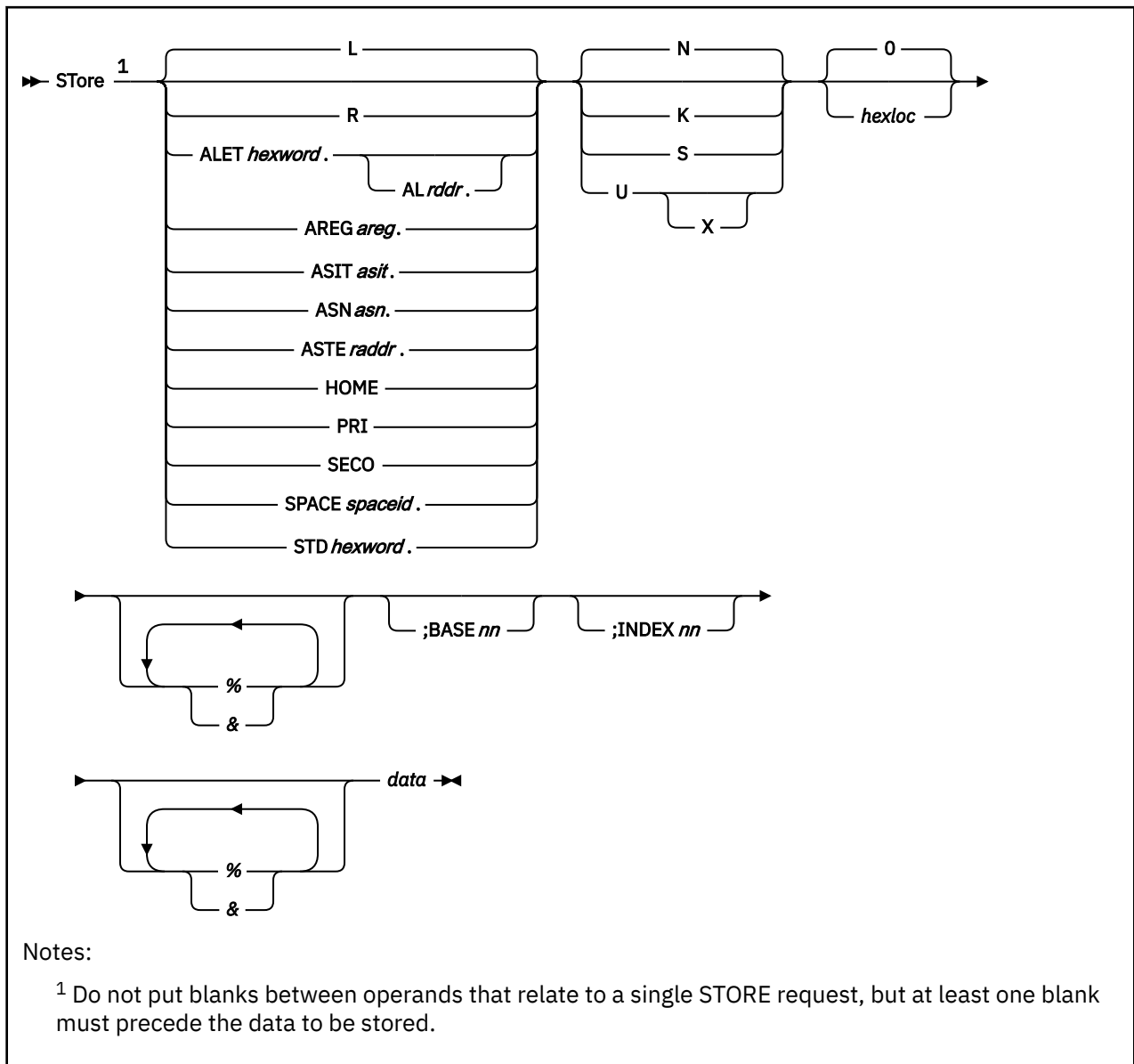
Store complete

indicates that the data has been stored.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP005E Invalid hexdata - *hexdata*
- HCP010E Invalid register - *register*
- HCP012E Invalid PSW - *psw*
- HCP026E Operand missing or invalid
- HCP033E Hexloc missing or invalid
- HCP163E STORE exceeds maximum register
- HCP260E {Command||*command* command} processing cannot complete due to errors.
- HCP657E The virtual machine's page zero is unavailable because it is damaged.
- HCP6025E Operand {STATUS|STORE STATUS|RESTART} is invalid because CPU *nn* is in check-stop state.
- HCP6150E An option has been specified that is not valid in {ESA/XC|z/XC} mode.
- HCP6525E Authorization request failed.

## STORE (Guest Storage - ESA/390)



### Authorization

Privilege Class: G

### Purpose

Use STORE (Guest Storage - ESA/390) to store data in the second-level or third-level storage of an ESA/390 virtual machine.

### Operands

**L**

stores the data in second-level storage (storage that appears real to your virtual machine) if the virtual machine is operating in DAT OFF mode, or stores the data in third-level storage (storage that appears virtual to your virtual machine) if the virtual machine is operating in DAT ON mode. This is the default.

The address space selected corresponds to the setting of the address space control field in the guest PSW and is the primary, secondary, or (if the virtual machine is an ESA virtual machine) home space. When the PSW indicates access register mode, the primary space is selected unless a nonzero *BASEnn* operand is also specified and the virtual machine is an ESA virtual machine, in which case an access register specified address space is selected.

**R**

stores the data in second-level storage (storage that appears real to your virtual machine).

**ALETheyword.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the specified access list entry token (ALET). The ALET is used in conjunction with guest access register translation tables to locate the guest address space to store data in.

**ALraddr.**

specifies an access list (AL) to be used in conjunction with the ALET to locate the guest address space. The guest real address of the access list may be 1 to 8 hexadecimal digits.

**AREGareg.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the contents of the specified access register. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The access list entry token (ALET) within the access register is used in conjunction with guest access register translation tables to locate the guest address space to store data in.

**ASITasit.**

stores the data in absolute storage in the CP-created address space identified by the specified address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

**ASNasn.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the specified address space number (ASN). The *asn* may be 1 to 4 hexadecimal digits. It used in conjunction with guest ASN-translation tables to locate a guest address space to store data in.

**ASTERaddr.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the segment table designation contained within the ASN second table entry (ASTE) located at guest real address *raddr*.

**HOME**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using guest control register 13.

**PRI**

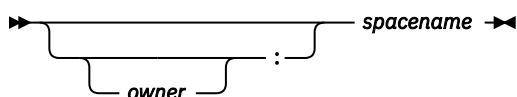
stores the data in third-level storage (storage that appears virtual to your virtual machine) using guest control register 1.

**SECO**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using guest control register 7.

**SPACEspaceid.**

stores the data in absolute storage in the CP-created address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the CP-created address space, and *spacename* is the 1-character to 24-character address space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

**STDhexword.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the specified guest segment table designation (STD).

**N**

stores the data (in hexadecimal) in successive fullword locations starting at the address specified by *hexloc*. The smallest group of hexadecimal values that can be stored is one fullword. This is the default.

If *hexloc* is not on a fullword boundary, the store operation begins on the next lower fullword boundary. If the value being stored is less than one fullword (8 hexadecimal digits) in length, it is right-adjusted in the word and is filled in with zeros. If two or more hexadecimal words are specified, they must be separated with one or more blanks.

**K**

stores the data in the storage key for the 4 KB page in which *hexloc* is located. Two hexadecimal digits are used to form the key for the page. If one digit is provided, it becomes the lower digit, and the upper digit is zero.

**S**

stores the data (in hexadecimal) without word alignment, starting at the address specified in *hexloc*. The shortest string that can be stored is 1 byte (2 hexadecimal digits).

If the string contains an odd number of characters, the last character is not stored; an error message is issued, and the function terminates. If you specify the S operand, the data you enter must be a string of 2 or more hexadecimal digits with no embedded blanks.

**U**

stores the data starting at the address specified by *hexloc*. The data is treated as an EBCDIC string by default; it is treated as an ASCII string if the X option is specified. The data is stored in byte segments with no word alignment.

**hexloc**

is the starting storage location that is to be altered. The address *hexloc* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The default is zero.

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is turned off) are used as the new address for the STORE command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the new address for the STORE command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

**BASEnn**

stores the data in guest storage using the specified address (*hexloc*) as a displacement from the address in the specified general register (*nn*). The general register may be a specified as a decimal number from 0 to 15 or as a hexadecimal number from 0 to F. *nn* indicates an AR/GR pair if all of the following are true:

- The L operand is specified.
- The virtual machine is an ESA virtual machine in ESA/390 mode (the default mode).
- The virtual CPU is in access register mode.



The specified access register is used, in the same manner as if the *AREGareg.* operand had been specified, in conjunction with the guest access-register-translation tables to locate the guest address space to store data in.

### **INDEXnn**

stores the data in guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

### **data**

is the data you want to place in storage. The operand takes on different meanings, depending on the type of STORE you are performing (N, K, S, or U). See the descriptions of each of these operands for the meaning of *data*.

## **Usage Notes**

1. An attempt to store at an address in a protected page, or to change the key at an address in a protected page, results in an error message. A protected page is a page of read-only storage.
2. The operands HOME, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* are valid only if the virtual machine is an ESA virtual machine.
3. When you use the operand *ALETheyword.ALraddr.*, the primary list bit in the ALET is ignored.
4. The operands L, PRI, SECO, HOME, *ASTERaddr.*, *STDhexword.*, *ASNasn.*, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* use the access register translation (ART), the ASN translation or the dynamic address translation (DAT) processes. These contain special case translation processes including the following:
  - ALET 0 always indicates the current primary address space designated by the STD in guest control register 1, regardless of the current contents of the dispatchable-unit access list.
  - ALET 1 always indicates the current secondary address space designated by the STD in guest control register 7, regardless of the current contents of the dispatchable-unit access list.
  - Access register 0 is always treated as containing the ALET 0. The contents of access register 0 are not used during ART processing.
  - General register 0 is always treated as containing the value 0 when used as the base or index register. The contents of GR 0 are never used.
5. The BASE operand is primarily intended for use in conjunction with the L operand. It allows a user to modify the storage the virtual machine uses in one less step than would otherwise be possible. Without the BASE operand, a user may need to display the contents of a general register, then add the displacement, and then enter the STORE command to modify storage the program is using. With the use of access registers, the BASE option allows the user to modify storage without needing to know whether the program is currently in access register mode. This is done automatically by CP.
6. If you use the *BASEnn* and *INDEXnn* operands together, the specified *hexloc* is used as a displacement from the address formed by the sum of the contents of the two specified registers.
 

The values in the registers specified by the *BASEnn* and *INDEXnn* operands are treated as 24-bit or 31-bit real addresses according to the addressing mode (A) bit in the virtual machine's PSW. That is, if the A bit is 1, the value is a 31-bit address. If the A bit is 0, the value is a 24-bit address. Address wrap-around is applied to the sum of the specified starting address and the contents of the two specified registers.
7. During STORE command processing, the authorization mechanisms used during normal address translation are not used.
8. Special authorization is required to use the SPACE or ASIT operand of this command to alter any address space other the host-primary address space of the virtual CPU. In order to use this command to alter storage in an address space, one of the following conditions must be satisfied:
  - Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.

## STORE (Guest Storage - ESA/390)

- If the address space is owned by another virtual machine, your virtual machine must have read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.

Authorization is granted on a space-by-space basis, so that you may be authorized to alter one address space of a virtual machine but may not be authorized to alter a different space.

9. The operands V and W used to place data in third-level storage from guest primary and secondary are being supported for compatibility. The PRI and SECO operands replace the V and W operands.
10. When the ASIT or SPACE operand is specified, the STORE request fails if the specified address space does not exist, or if you are not authorized to alter the address space. The current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
11. The STORE command requires a segment table designation (STD) to alter third-level storage (storage that appears virtual to your virtual machine). It may receive the STD as an operand, find it in a control register, or use the access-register translation (ART) process or the address space number (ASN) translation process to obtain the STD. The access-register translation and ASN translation processes may fail because of one of several program exception conditions. If the translation does not succeed, the current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a response indicating the nature of the exception condition.
12. If you use an underscore, it must be followed by 8 hexadecimal digits.
13. If you issue STORE with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the STORE command is rejected and error message HCPSTT6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CClear option
  - SET MACHine (to a different machine architecture)

## Responses

Response 1:

```
Store complete
```

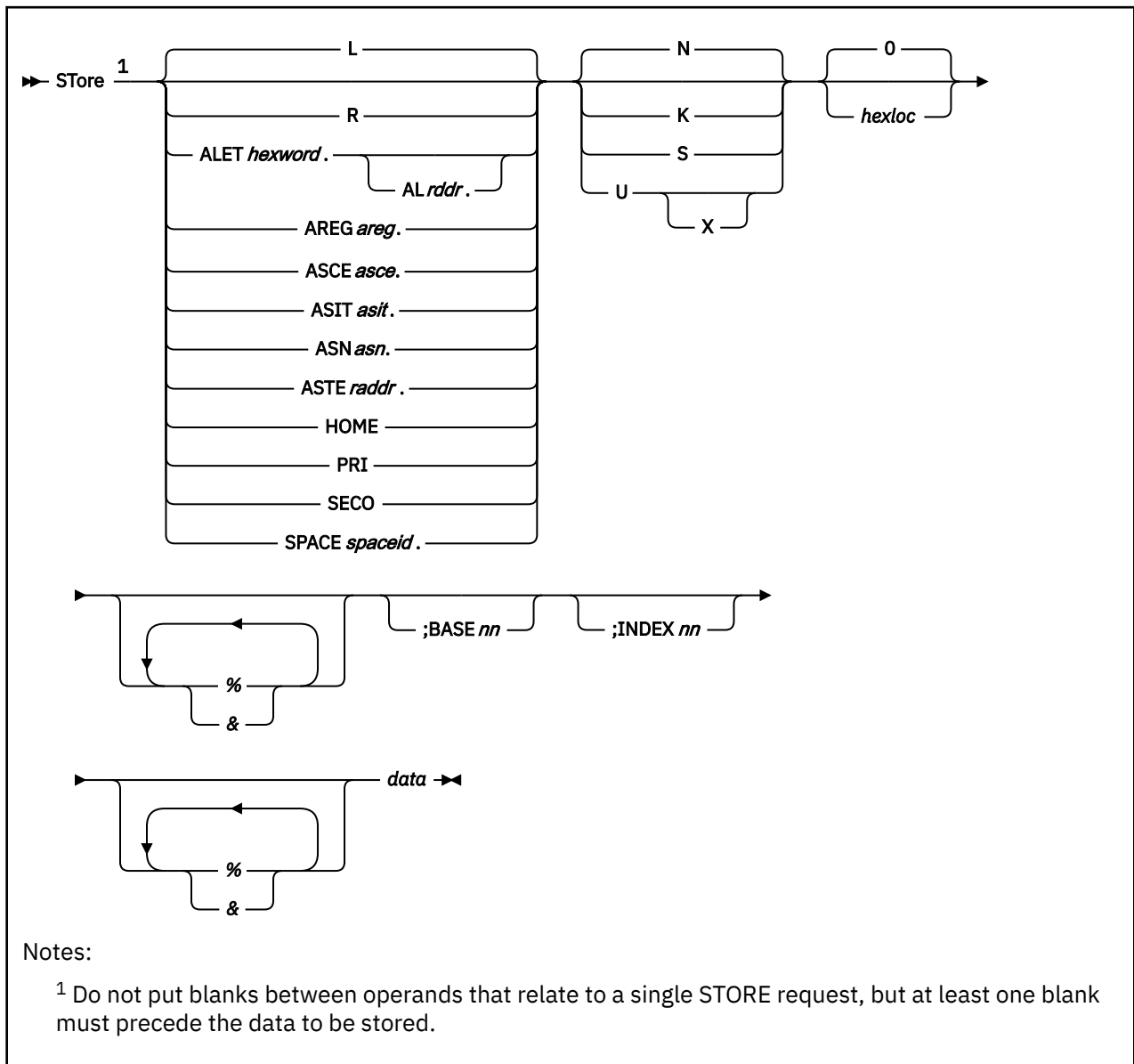
is issued when the command completes successfully.

## Messages

- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*
- HCP005E Invalid hexdata - *hexdata*
- HCP010E Invalid register - *register*
- HCP012E Invalid PSW - *psw*
- HCP026E Operand missing or invalid
- HCP033E Hexloc missing or invalid
- HCP160E Hexloc *hexloc* exceeds storage
- HCP163E STORE exceeds maximum register
- HCP164E Hexloc *hexloc* non-addressable storage
- HCP260E *{Command|command command}* processing cannot complete due to errors.
- HCP261E *key* is an invalid storage key.

- HCP422E The same option was specified twice.
- HCP657E The virtual machine's page zero is unavailable because it is damaged.
- HCP1001E An operand is missing for *option*.
- HCP2611E {Data|Key} cannot be stored into a read-only page.
- HCP6025E Operand {STATUS|STORE STATUS|RESTART} is invalid because CPU *nn* is in check-stop state.
- HCP6150E An option has been specified that is not valid in ESA/390 mode.
- HCP6151E Storage {operand} is not addressable because of { a protection | an addressing | a segment translation | a page translation | a translation specification | an ASN translation | an AFX translation | an ASX translation | an ALET translation | an ALEN translation | an ALE sequence | an ASTE validity | an ASTE sequence } exception.
- HCP6525E Authorization request failed.

## STORE (Guest Storage - z/Architecture)



### Authorization

Privilege Class: G

### Purpose

Use STORE (Guest Storage - z/Architecture) to store data in the second-level or third-level storage of a virtual machine in z/Architecture mode.

### Operands

L

stores the data in second-level storage (storage that appears real to your virtual machine) if the virtual machine is operating in DAT OFF mode, or stores the data in third-level storage (storage that appears virtual to your virtual machine) if the virtual machine is operating in DAT ON mode. This is the default.

The address space selected corresponds to the setting of the address space control field in the guest PSW and is the primary, secondary, or (if the virtual machine is an ESA virtual machine) home space. When the PSW indicates access register mode, the primary space is selected unless a nonzero *BASEnn* operand is also specified and the virtual machine is an ESA virtual machine, in which case an access register specified address space is selected.

**R**

stores your data in second-level storage (storage that appears real to your virtual machine).

**ALEThexword.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the specified access list entry token (ALET). The ALET is used in conjunction with guest access register translation tables to locate the guest address space to store data in.

**ALraddr.**

specifies an access list (AL) to be used in conjunction with the ALET to locate the guest address space. The guest real address of the access list may be 1 to 8 hexadecimal digits.

**AREGareg.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the contents of the specified access register. The *areg* may be specified as 1 hexadecimal or 2 decimal digits. The ALET within the access register is used in conjunction with guest access register translation tables to locate the guest address space to store data in.

**ASCEasce.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the specified address-space-control element (ASCE). The *asce* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability.

**ASITasit.**

stores the data in absolute storage in the CP-created address space identified by the specified address space identification token (ASIT). The *asit* may be 1 to 16 hexadecimal digits; leading zeros are optional.

**ASNasn.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the specified address space number (ASN). The *asn* may be 1 to 4 hexadecimal digits. It is used in conjunction with guest ASN-translation tables to locate the guest address space to store data in.

**ASTERaddr.**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using the segment table designation contained within the ASN second table entry (ASTE) located at guest real address *raddr*.

**HOME**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using guest control register 13.

**PRI**

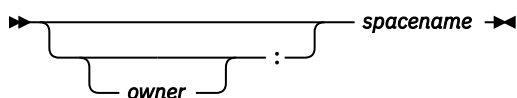
stores the data in third-level storage (storage that appears virtual to your virtual machine) using guest control register 1.

**SECO**

stores the data in third-level storage (storage that appears virtual to your virtual machine) using guest control register 7.

**SPACEspaceid.**

stores the data in absolute storage in the CP-created address space identified by *spaceid*. The *spaceid* is a 1-character to 33-character string of the form:



## STORE (Guest Storage - z/Architecture)

where *owner* is an optional 1-character to 8-character user ID of the virtual machine that owns the CP-created address space, and *spacename* is the 1-character to 24-character address space name assigned to the address space.

If *owner* is not specified, your virtual machine is assumed to be the owner of the address space. The *spacename* 'BASE' is preassigned by CP to be the name of your virtual machine's host-primary address space.

### N

stores the data (in hexadecimal) in successive fullword locations starting at the address specified by *hexloc*. The smallest group of hexadecimal values that can be stored is one fullword. This is the default.

If *hexloc* is not on a fullword boundary, the store operation begins on the next lower fullword boundary. If the value being stored is less than one fullword (8 hexadecimal digits) in length, it is right-adjusted in the word and is filled in with zeros. If two or more hexadecimal words are specified, they must be separated with one or more blanks.

### K

stores the data in the storage key for the 4 KB page in which *hexloc* is located. Two hexadecimal digits are used to form the key for the page. If one digit is provided, it becomes the lower digit, and the upper digit is zero.

### S

stores the data (in hexadecimal) without word alignment, starting at the address specified in *hexloc*. The shortest string that can be stored is 1 byte (2 hexadecimal digits).

If the string contains an odd number of characters, the last character is not stored; an error message is issued, and the function terminates. If you specify the S operand, the data you enter must be a string of two or more hexadecimal digits with no embedded blanks.

### U

stores the data starting at the address specified by *hexloc*. The data is treated as an EBCDIC string by default; it is treated as an ASCII string if the X option is specified. The data is stored in byte segments with no word alignment.

### *hexloc*

is the starting storage location that is to be altered. The address *hexloc* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The default is zero.

### %

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is set to 0) are used as the address for the STORE command.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

### &

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the address for the STORE command. A maximum of 16 indirection characters can be specified.

When specified following the INDEX and BASE operands, the indirect address is determined after BASE and INDEX are applied.

### **BASEnn**

stores the data in guest storage using the specified address (*hexloc*) as a displacement from the address in the specified general register (*nn*). The general register may be specified as a decimal number from 0 to 15 or as a hexadecimal number from 0 to F. *nn* specifies an AR/GR pair if all of the following are true:

- The L operand is specified.
- The virtual machine is an ESA virtual machine in z/Architecture mode.

**Note:** An ESA virtual machine simulates ESA/390 architecture when initially defined. However, a guest may have the capability to switch the virtual machine from ESA/390 mode to z/Architecture mode.

- The virtual CPU is in access register mode.

The specified access register is used, in the same manner as if the *AREGareg.* operand had been specified, in conjunction with the guest access-register-translation tables to locate the guest address space to store data in.

### INDEXnn

stores the data in guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified register. The specified register may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F.

### data

is the data you want to place in storage. The operand takes on different meanings, depending on the type of STORE you are performing (N, K, S, or U). See the descriptions of each of these operands for the meaning of *data*.

## Usage Notes

1. An attempt to store at an address in a protected page, or to change the key at an address in a protected page, results in an error message. A protected page is a page of read-only storage.
2. The operands HOME, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* are valid only if the the virtual machine is an ESA virtual machine.
3. When you use the operand *ALETheyword.ALraddr.*, the primary list bit in the ALET is ignored.
4. The operands L, PRI, SECO, HOME, *ASTERaddr.*, *ASCEasce.*, *ASNasn.*, *AREGareg.*, *ALETheyword.*, and *ALETheyword.ALraddr.* use the access register translation (ART), the ASN translation or the dynamic address translation (DAT) processes. These contain special case translation processes including the following:
  - ALET 0 always indicates the current primary address space designated by the ASCE in guest control register 1, regardless of the current contents of the dispatchable-unit access list.
  - ALET 1 always indicates the current secondary address space designated by the ASCE in guest control register 7, regardless of the current contents of the dispatchable-unit access list.
  - Access register 0 is always treated as containing the ALET 0. The contents of access register 0 are not used during ART processing.
  - General register 0 is always treated as containing the value 0 when used as the base or index register. The contents of GR 0 are never used.
5. The BASE operand is primarily intended for use in conjunction with the L operand. It allows a user to modify the storage the virtual machine uses in one less step than would otherwise be possible. Without the BASE operand, a user may need to display the contents of a general register, then add the displacement, and then enter the STORE command to modify storage the program is using. With the use of access registers, the BASE option allows the user to modify storage without needing to know whether the program is currently in access register mode. This is done automatically by CP.
6. If you use the *BASEnn* and *INDEXnn* operands together, the specified *hexloc* is used as a displacement from the address formed by the sum of the contents of the two specified registers. The values in the registers specified by the *BASEnn* and *INDEXnn* operands are treated as 24-bit, 31-bit, or 64-bit real addresses according to the addressing mode bits (EA and BA - bits 31 and 32) in the virtual machine's PSW. The addressing mode is determined as follows:

Bit 31	Bit 32	Addressing Mode
0	0	24-bit
0	1	31-bit
1	1	64-bit

## STORE (Guest Storage - z/Architecture)

7. During STORE command processing, the authorization mechanisms used during normal address translation are not used.
  8. Special authorization is required to use the SPACE or ASIT operand of this command to alter any address space other the host-primary address space of the virtual CPU. In order to use this command to alter storage in an address space, one of the following conditions must be satisfied:
    - Your virtual machine must be the owner of the address space. Your virtual machine is the owner of its host-primary address space and of any address space that it creates using the CREATE function of the ADRSPACE macro.
    - If the address space is owned by another virtual machine, your virtual machine must have read-write authorization to the address space. The owner of the address space can grant authorization to your virtual machine using the PERMIT function of the ADRSPACE macro.
- Authorization is granted on a space-by-space basis, so that you may be authorized to alter one address space of a virtual machine but may not be authorized to alter a different space.
9. The operands V and W used to place data in third-level storage from guest primary and secondary are being supported for compatibility. The PRI and SECO operands replace the V and W operands.
  10. When the ASIT or SPACE operand is specified, the STORE request fails if the specified address space does not exist, or if you are not authorized to alter the address space. The current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a message indicating the nature of the error.
  11. The STORE command requires a segment table designation (ASCE) to alter third-level storage (storage that appears virtual to your virtual machine). It may receive the ASCE as an operand, find it in a control register, or use the access-register translation (ART) process or the address space number (ASN) translation process to obtain the ASCE. The access-register translation and ASN translation processes may fail because of one of several program exception conditions. If the translation does not succeed, the current request is rejected, and processing of the command line is terminated. The rejected operands are displayed along with a response indicating the nature of the exception condition.
  12. If you use an underscore, it must be followed by 8 hexadecimal digits.
  13. If you issue STORE with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the STORE command is rejected and error message HCPSTT6156E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
    - SYSTEM CLEAR
    - IPL by NSS name
    - IPL by device with the CClear option
    - SET MACHine (to a different machine architecture)

## Responses

Response 1:

```
Store complete
```

is issued when the command completes successfully.

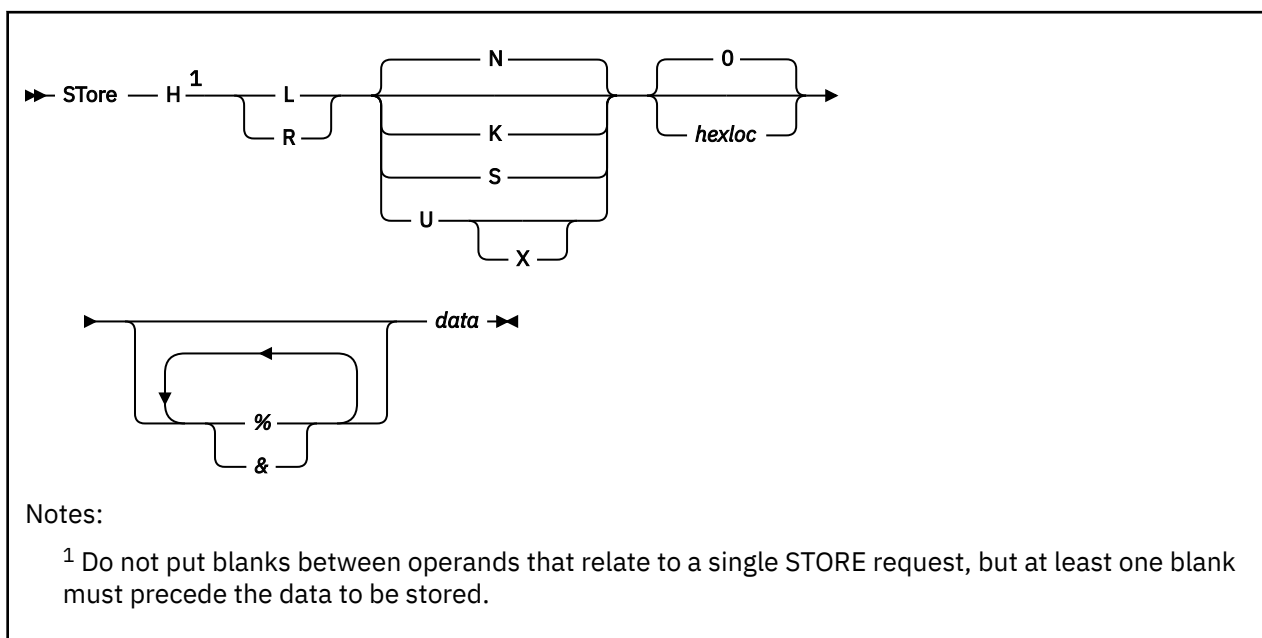
## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP005E Invalid hexdata - *hexdata*
- HCP010E Invalid register - *register*



- HCP012E Invalid PSW - *psw*
- HCP026E Operand missing or invalid
- HCP033E Hexloc missing or invalid
- HCP160E Hexloc *hexloc* exceeds storage
- HCP163E STORE exceeds maximum register
- HCP164E Hexloc *hexloc* non-addressable storage [- block volatility exception]
- HCP260E {Command|*command* command} processing cannot complete due to errors.
- HCP261E *key* is an invalid storage key.
- HCP422E The same option was specified twice.
- HCP657E The virtual machine's page zero is unavailable because it is damaged.
- HCP1001E An operand is missing for *option*.
- HCP2611E {Data|Key} cannot be stored into a read-only page.
- HCP6025E Operand {STATUS|STORE STATUS|RESTART} is invalid because CPU *nn* is in check-stop state.
- HCP6150E An option has been specified that is not valid in z/Architecture mode.
- HCP6151E Storage {operand} is not addressable because of { a protection | an addressing | a segment translation | a page translation | a translation specification | an ASN translation | an AFX translation | an ASX translation | an ALET translation | an ALEN translation | an ALE sequence | an ASTE validity | an ASTE sequence } exception.
- HCP6525E Authorization request failed.

## STORE (Host Storage)



### Authorization

Privilege Class: C

### Purpose

Use STORE (Host Storage) to store data into host (first-level) storage.



**Attention:** Careless storage alterations can cause abnormal terminations of your system as well as security and integrity problems.

### Operands

**L**

specifies that the data is to be stored into host logical storage.

**R**

specifies that the data is to be stored into host real storage.

**N**

stores the specified (hexadecimal) data in successive fullword locations starting at the address specified by *hexloc*. The smallest group of hexadecimal values that can be stored is one fullword. This is the default.

If *hexloc* is not on a fullword boundary, the store operation begins on the next lower fullword boundary. If the value being stored is less than one fullword (four bytes) in length, it is right-adjusted in the word and is filled in with zeros. If two or more hexadecimal words are specified, they must be separated with one or more blanks.

**K**

stores the data in the storage key for the 4 KB page or frame in which *hexloc* is located. Two hexadecimal digits form the storage key. If one digit is provided, it becomes the lower digit, and the upper digit is 0.

If you are storing data into host logical storage, the storage key being changed is for the backing frame in host real storage at that moment.

**S**

stores the specified hexadecimal data without word alignment starting at the address specified in *hexloc*. The shortest string that can be stored is 1 byte (2 hexadecimal digits). If the string contains an odd number of characters, the last character is not stored; an error message is issued, and the function terminates. If you specify the S operand, the data you enter must be a string of 2 or more hexadecimal digits with no embedded blanks.

**U**

stores the specified data starting at the address specified by *hexloc*. By default the data is treated as EBCDIC; if the X option is specified, the data is treated as ASCII. The data is stored in byte segments with no word alignment.

**hexloc**

is the starting storage location that is to be altered. The address *hexloc* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability; leading zeros are optional. The default is zero.

This address must be within the bounds of addressable storage (see usage note “3” on page 1899).

**%**

indicates 31-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 31 bits at the specified address (the high-order bit is set to 0) are used as the address for the STORE command.

**&**

indicates 64-bit address indirection. Indirection means that the contents of the specified address are used instead of the address itself. In this case, the 64 bits at the specified address are used as the address for the STORE command.

**data**

is the data you want to place in storage. The operand takes on different meanings, depending on the type of STORE you are performing (N, S, K, or U). See the descriptions of each of these operands for the meaning of *data*.

**Usage Notes**

1. In a multiprocessor environment, each processor has a prefix register that it uses to relocate addresses between 0 and X'1FFF' (an 8 KB area known as the prefix area) to two other contiguous frames in real storage. The prefix register enables each processor to use a different prefix area to avoid conflict with other processors for such activity as interrupt code recording. Thus, the range 0 through X'1FFF' refers to different areas of storage, depending on which processor generates the address. All references to first-level storage are handled as if they were made on the IPL processor.
2. If an external security manager is installed on your system, you may not be authorized to enter this command. For additional information, contact your security administrator.
3. For host logical storage (the system execution space), addressable storage includes the system execution area, which resides below 2 GB, plus the system frame table and the system execution space page management table, which reside above 2 GB. To display the storage address ranges used for the system execution space, issue the QUERY SXSSTORAGE command. The storage locations that you alter with the STORE command must be within the same address range of the system execution space.  
  
For host real storage, addressable storage is the lesser of CP's configured storage size (if specified at system generation) and the real machine storage size. To display the size of host real storage, issue the QUERY STORAGE command.
4. When storing data into host logical storage, if you specify an address that is not backed with a real storage frame, error message HCP6151E is issued, even if the address is included in the ranges of in-use and usable addresses displayed by the QUERY SXSSTORAGE command.
5. If you use an underscore, it must be followed by 8 hexadecimal digits.

## STORE (Host Storage)

### Responses

Response 1:

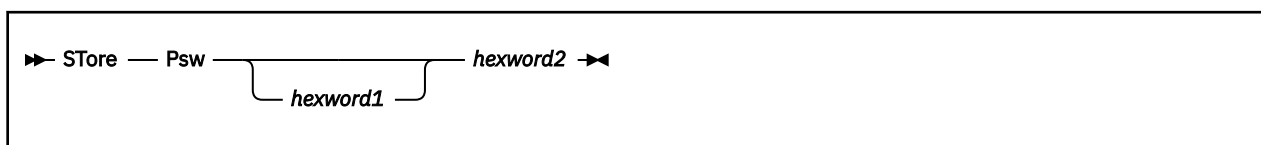
```
Store complete
```

is issued when the command completes successfully.

### Messages

- HCP160E Hexloc *hexloc* exceeds storage
- HCP164E Hexloc *hexloc* non-addressable storage
- HCP6065E STORE HOST requires R (real) or L (logical) address type

## STORE PSW



### Authorization

Privilege Class: G

### Purpose

Use STORE PSW to alter the program status word (PSW).

### Operands

#### PSW

For an ESA/390 or ESA/XC guest:

This alters all or part of the PSW with the data you specify in *hexword1* and *hexword2*. If you specify only *hexword2*, it is stored in the second word of your virtual machine's current PSW. If you specify both *hexword1* and *hexword2*, *hexword1* is stored in the first word of your virtual machine's current PSW, and *hexword2* is stored into the address portion of the PSW.

For a z/Architecture or z/XC guest:

This is allowed only if a SET PSWTRANS ALL/STORE command has previously been issued for the virtual machine. If the appropriate SET PSWTRANS command has been issued, the hexwords are converted from an ESA/390-format PSW into a z/Architecture-format PSW as indicated below:

If only *hexword2* is specified, the z/Architecture PSW is stored as follows:

- bit 32 = *hexword2* bit 0
- bits 33-96 = 0
- bits 97-127 = *hexword2* bits 1-31
- rest of bits remain unchanged

If you specify *hexword1 hexword2*, the z/Architecture PSW is stored as follows:

- bits 0-31 = *hexword1* (bit 12 is set to 0)
- bit 32 = 1 if *hexword1* bit 31 is 1 otherwise, this bit will be set to *hexword2* bit 0
- bits 33-96 = 0
- bits 97-127 = *hexword2* bits 1-31

#### **hexword...**

identifies the data you want to store in the PSW. If the data is less than eight digits (four bytes), it is right-adjusted and padded with leading zeros.

### Usage Notes

1. To specify a 128-bit PSW, use the STORE PSWG command.

### Responses

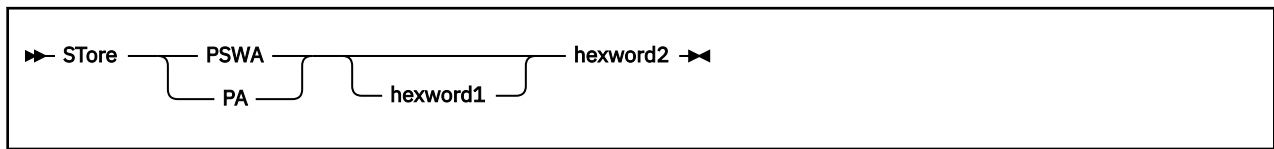
Response 1:

## STORE PSW

Store complete

is issued when the command completes successfully.

## STORE PSWA



### Authorization

Privilege Class: G

### Purpose

Use STORE PSWA to alter the address portion only of the current PSW.

### Operands

#### PSWA

For ESA/390 or ESA/XC guest:

One *hexword* of data (*hexword2*) is accepted as the new address. The addressing mode bit in the current PSW remains unchanged.

For a z/Architecture or z/XC guest:

One or two *hexwords* are accepted as the new address. The *hexword(s)* are stored in the address portion of the PSW, the rest of the PSW remains unchanged. If only one *hexword* is specified it is stored in the right-most word of the address portion of the PSW, the left-most word of the address is set to 0.

#### *hexword...*

identifies the data you want to store in the address portion of the PSW. If the data is less than eight digits (four bytes), it is right-adjusted and padded with leading zeros.

### Messages

- HCPSTC026E Operand missing or invalid
- HCPSTC6149E The address specified is not valid in the current addressing mode

## STORE PSWG



### Authorization

Privilege Class: G

### Purpose

Use STORE PSWG to alter the program status word for a z/Architecture or z/XC virtual machine.

### Operands

#### PSWG

#### PG

alters all or part of the PSW with the data you specify in *flagw1*, *flagw2*, *addrw1*, and *addrw2*.

If you specify only *addrw2*, or *addrw2* and *addrw1* the contents are stored right-justified, in the address portion of your virtual machine's PSW. If *flagw1* and *flagw2* are also specified, the complete PSW is stored. Specifying three parameters results in an error, for example *flagw2 addrw1 addrw2*.

Data less than eight digits (four bytes), is right-adjusted and padded with leading zeros.

#### *flagw1*

the data (in hexadecimal) to be stored in the first word of the z/Architecture PSW. If omitted, this portion of the PSW is left unchanged.

#### *flagw2*

the data (in hexadecimal) to be stored in the second word of the z/Architecture PSW. If omitted, this portion of the PSW is left unchanged.

#### *addrw1*

the data (in hexadecimal) to be stored in the third word of the z/Architecture PSW. If omitted, this portion of the PSW is set to 0.

#### *addrw2*

the data (in hexadecimal) to be stored in the fourth word of the z/Architecture PSW. This parameter is required.

### Usage Notes

To store a PSW for an ESA/390 or ESA/XC virtual machine, use the STORE PSW command.

### Responses

Response 1:

```
Store complete
```

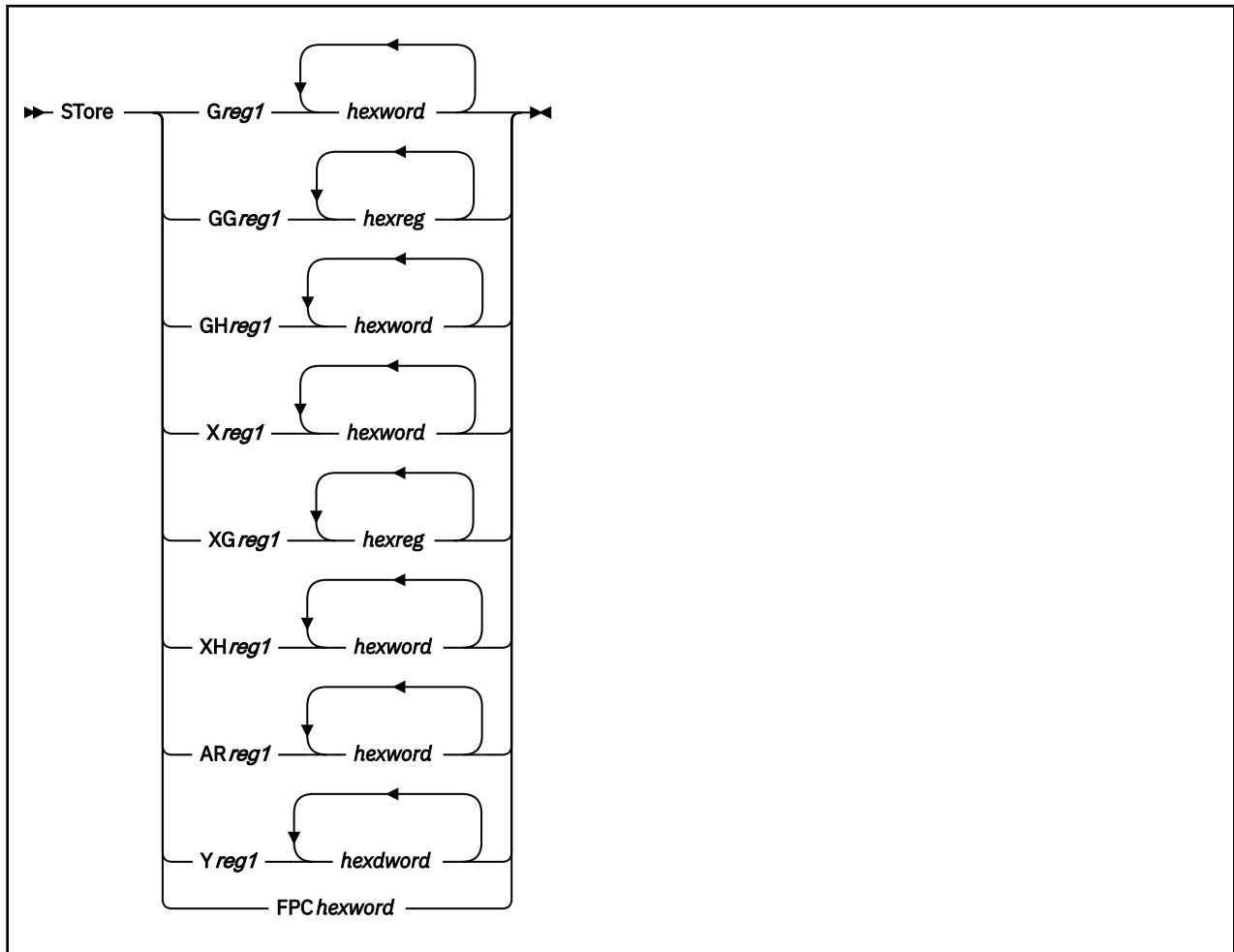
is issued when the command completes successfully.

### Messages

- HCPSTC026E Operand missing or invalid



## STORE (Registers)



### Authorization

Privilege Class: G

### Purpose

Use STORE (Registers) to store data into the following:

- General-purpose registers
- Control registers
- Access registers
- Floating-point registers
- Floating-point control register.

### Operands

#### *Greg1*

G indicates a full general-purpose register for an ESA/390 or ESA/XC guest, or the low-order half of the register for a z/Architecture or z/XC guest. *reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, general register whose contents are to be assigned a value.

## STORE (Registers)

### **GGreg1**

G indicates a full general-purpose register (32 bits for an ESA/390 or ESA/XC guest, 64 bits for a z/Architecture or z/XC guest). *reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, general register whose contents are to be assigned a value.

### **GHreg1**

GH indicates the high-order half of a general-purpose register for a z/Architecture or z/XC guest. This option is not valid for an ESA/390 or ESA/XC guest. *reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, general register whose contents are to be assigned a value.

### **Xreg1**

X indicates a full control register for an ESA/390 or ESA/XC guest or the low-order half of the register for a z/Architecture or z/XC guest. *reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, control register that is to be assigned a value.

### **XGreg1**

XG indicates a full control register (32 bits for an ESA/390 or ESA/XC guest, 64 bits for a z/Architecture or z/XC guest). *reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, control register that is to be assigned a value.

### **XHreg1**

XH indicates the high-order half of a control register for a z/Architecture or z/XC guest. This option is not valid for an ESA/390 or ESA/XC guest. *reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, control register whose contents are to be assigned a value.

### **ARreg1**

AR indicates access registers. *reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, access register whose contents are to be assigned a value.

### **Yreg1**

Y indicates floating-point registers. *reg1* is a decimal number from 0 to 15, or a hexadecimal number from 0 to F, representing the first, or only, floating-point register whose contents are to be assigned a value.

### **FPC**

is the floating-point control register and is available only if floating-point extensions are installed and enabled for the virtual machine issuing the STORE command.

Some bits in the floating-point control register are reserved and cannot be altered (see the Principles of Operation for your processor). The data specified by the *hexword* for the FPC operand is checked for validity on this machine to ensure that none of the reserved bits are altered. If the *hexword* data for the FPC operand attempts to change any of the reserved bits, the message *HCP005E Invalid hexdata* is displayed and the floating-point control register is not changed.

### **hexword**

each *hexword* represents up to eight hexadecimal digits for G, GH, X, XH, and AR register stores. Only one hexword is allowed for FPC. If the value being stored is fewer than the eight digits, it is right-adjusted in the register, and the high-order bytes of the register are filled with zeros. If two or more values are specified, they must be separated by one or more blanks. The data is stored in successive registers.

### **hexdword**

each *hexdword* represents up to 16 hexadecimal digits for Y register stores. If the value being stored is fewer than 16 digits, it is left-adjusted in the register and is padded on the right with zeros. If two or more values are specified, they must be separated by one or more blanks. The data is stored in successive registers.

### **hexreg**

each *hexreg* represents up to eight hexadecimal digits for GG and XG stores for an ESA/390 or ESA/XC guest, or 16 hexadecimal digits, separated with an optional underscore, for a z/Architecture or z/XC guest. If the value being stored is less than the maximum number of hexadecimal digits allowed, it is right-adjusted in the register and is padded on the left with zeros. If two or more values are specified, they must be separated by one or more blanks. The data is stored in successive registers.

## Usage Notes

1. The G and X operands are valid for an ESA/390, ESA/XC, z/Architecture, or z/XC guest. For an ESA/390 or ESA/XC guest, these operands refer to the full four-byte register. For a z/Architecture or z/XC guest, these operands refer to the low-order four bytes of the full eight-byte register; the high-order half of the register remains unchanged.
2. The GG and XG operands are valid for an ESA/390, ESA/XC, z/Architecture, or z/XC guest. For an ESA/390 or ESA/XC guest, these operands refer to the full four-byte register. For a z/Architecture or z/XC guest, these operands refer to the full eight-byte register.
3. The GH and XH operands are valid only for a z/Architecture or z/XC guest. These operands refer to the high-order four bytes of the full eight-byte register.
4. When part of an SSI cluster, the validity of the FPC operand depends on the relocation domain of the issuing user, and may differ among users on the same system when they belong to different relocation domains.

## Responses

Response 1:

```
Store complete
```

is issued when the command completes successfully.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP005E Invalid hexdata - *hexdata*
- HCP010E Invalid register - [*register*] [*not specified.*]
- HCP012E Invalid PSW - *psw*
- HCP026E Operand missing or invalid
- HCP033E Hexloc missing or invalid
- HCP163E STORE exceeds maximum register
- HCP260E {Command|*command* command} processing cannot complete due to errors.
- HCP657E The virtual machine's page zero is unavailable because it is damaged.
- HCP6025E Operand {STATUS|STORE STATUS|RESTART} is invalid because CPU *nn* is in check-stop state.
- HCP6150E An option has been specified that is not valid in {ESA/390|ESA/XC|z/Architecture|z/XC} mode.
- HCP6525E Authorization request failed.

## STORE STATUS

► Store — STATUS ◄

### Authorization

Privilege Class: G

### Purpose

Use STORE STATUS to store selected virtual machine data in certain low storage locations, simulating the hardware store status facility. These locations are permanently assigned locations in guest absolute storage.

The STATUS operand causes the following data to be stored in low storage at the guest absolute addresses shown:

Address		Length	Content
Dec	Hex		
163	A3	1	Architectural-mode id (X'00')
212	D4	4	Extended save area address. See note below.
216	D8	8	CPU timer
224	E0	8	Clock comparator
256	100	8	Current PSW
264	108	4	Prefix register
288	120	64	Access registers 0 through 15
352	160	32	Floating-point registers 0, 2, 4, 6
384	180	64	General registers 0 through 15
448	1C0	64	Control registers 0 through 15

**Note:** The extended save area address is used if it is provided. When the extended save area is available, the virtual machine's floating-point registers 0 through 15 and floating-point control register are stored there.

Address		Length	Content
Dec	Hex		
0163	00A3	1	Architectural-mode id (X'01')
4608	1200	128	Floating-point registers 0 through 15
4736	1280	128	General registers 0 through 15
4864	1300	16	Current PSW
4888	1318	4	Prefix register

Table 44. z/Architecture mode guest (continued)

Address		Length	Content
Dec	Hex		
4892	131C	4	Floating-point control register
4900	1324	4	TOD programmable register
4904	1328	8	CPU timer
4913	1331	7	Clock comparator
4920	1338	8	Breaking-event-address register
4928	1340	64	Access registers 0 through 15
4992	1380	128	Control registers 0 through 15

### Usage Notes

1. The STORE STATUS command stores data in the logout area.
2. The STORE STATUS command stores CPU timer and clock comparator values that may then be displayed on your console through the DISPLAY command. This procedure is one way to get timer information on your display.

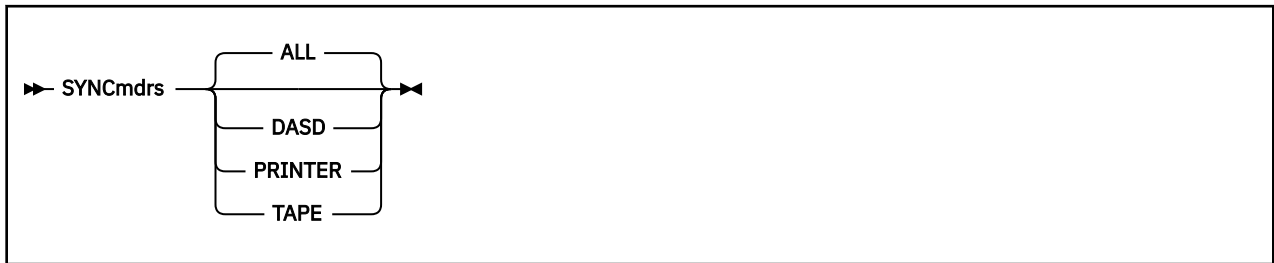
### Responses

Response 1:

```
Store complete
```

is issued when the command completes successfully.

## SYNCDMS



### Authorization

Privilege Class: A, B, F

### Purpose

Use the SYNCDMS command to retrieve Miscellaneous Data Records (MDRs) from specified devices. MDRs contain error and usage data from buffered control units or communications controllers. The records are logged in the VM error recording area.

### Operands

#### DASD

will capture all MDRs for all DASDs.

#### TAPE

will capture all MDRs for 3480, 3490, and 3590 tape devices.

#### PRINTER

will capture all MDRs for all 3800 printer devices.

#### ALL

will capture all MDRs for all devices. If no operand is specified, ALL is the default.

### Usage Notes

- Events that trigger automatic retrieval of MDRs usually occur infrequently. Some of these events are:
  - overflow of statistical counters in a buffered control unit
  - DASD volume demounts
  - VARY OFFLINE commands.

The SYNCDMS command lets you retrieve MDR records more frequently.

- SYNCDMS is performed synchronously.
- SYNCDMS does not get data from offline devices.

### Responses

Response 1:

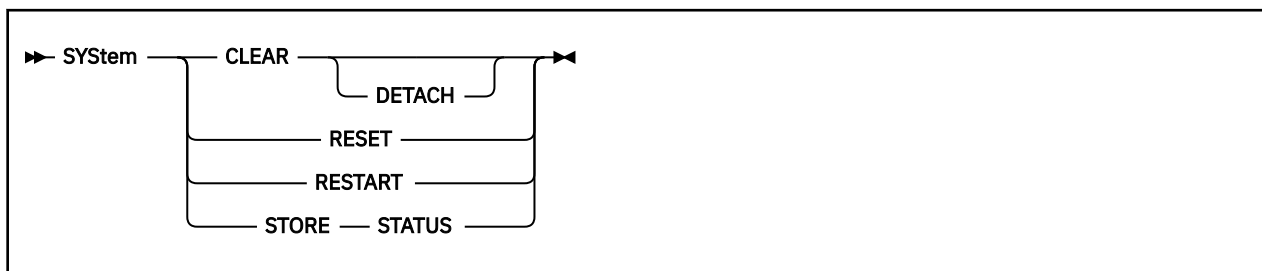
```
COMMAND COMPLETE
```

indicates that the command completed successfully.

### Messages

- HCP026E Operand missing or invalid

## SYSTEM



### Authorization

Privilege Class: G

### Purpose

Use SYSTEM to clear storage and to simulate the actions of RESET, RESTART, and STORE STATUS functions from a real console, as specified in [z/Architecture Principles of Operation](#).

### Operands

#### CLEAR

performs a system-reset-clear function on your virtual machine and places all processors in your configuration in a hard stop.

#### DETACH

detaches storage element 1, if it exists, before performing the system-reset-clear function. If element 1 does not exist, execution proceeds without error. The remaining assigned storage is cleared to zeros.

#### RESET

performs a system-reset-normal function on your virtual machine and places all processors in your virtual machine's configuration in a hard stop.

#### RESTART

simulates the hardware system RESTART function. If the virtual machine is in z/Architecture mode, this stores the current PSW into the quadword at guest real location 288 (120 hex) and loads the quadword from location 416 (1A0 hex) as the new PSW. Otherwise, it stores the current PSW into the doubleword at guest real location 8 and loads the doubleword from location 0 as the new PSW. Pending interruptions remain unaffected.

#### STORE STATUS

stores selected virtual machine data in certain low storage locations of the virtual machine, simulating the hardware store status facility. These locations are permanently assigned locations in guest absolute storage.

### Usage Notes

1. When the SYSTEM CLEAR command or the SYSTEM RESET command is entered from the virtual machine console as the first or only command on a CP command line (that is, while the terminal mode is CP, or CP READ is displayed or the command line begins with #CP), VM attempts to break out of certain wait conditions that might exist. Thus the SYSTEM CLEAR command or the SYSTEM RESET command may be entered to cause breakout of certain virtual machine wait conditions.

The breakout attempt occurs when the SYSTEM CLEAR command or the SYSTEM RESET command is entered, not when the command is processed. Command errors do not prevent the breakout attempt. In some cases, a breakout resets the virtual machine.

2. The RESET function and the CLEAR operands leave the virtual machine in a stopped state.

## SYSTEM

3. To resume operation after issuing the SYSTEM command with RESET specified, either store a PSW and enter BEGIN or enter BEGIN with a hexadecimal storage location specified. When doing either of these, be sure the wait state bit in the PSW is turned off before issuing BEGIN. The virtual machine automatically restarts at the location specified in the new PSW (which is loaded from the doubleword at location 0) after the SYSTEM RESTART command is processed.
4. The SYSTEM CLEAR or SYSTEM RESET command clears a virtual check-stop condition on the virtual processors.
5. The SYSTEM STORE STATUS command stores data in the logout area. It stores CPU timer and clock comparator values that may then be displayed on your display through the DISPLAY command. This procedure is one way to get timer information on your display.
6. You can use the CPU command to cause a restart interruption to occur on the virtual CPU of your choice. For example, "CPU 3 SYSTEM RESTART" generates the interruption on CPU 3.
7. SYSTEM RESTART cannot be used to resume execution on secondary processors that are not operational. In order for a secondary CPU to be started, the virtual machine must be in z/Architecture mode and the guest operating system must support secondary processors.
8. If the virtual storage configuration was defined with reserved or standby storage, then dynamic storage reconfiguration (DSR) is supported. Otherwise, DSR is not supported. If DSR is supported, then virtual storage is comprised of storage element 0 and storage element 1. Element 0 is always attached. You can use the DETACH option with SYSTEM CLEAR to detach storage element 1 before the system-reset-clear function is performed.
9. The SYSTEM RESET and SYSTEM CLEAR commands will fail if issued when VMRELOCATE is in progress.

### Responses

No response is issued by SYSTEM RESTART. The virtual machine resumes execution at the address in the virtual PSW loaded from virtual storage location 0.

Response 1:

```
STORAGE CLEARED - SYSTEM RESET
```

is issued by the SYSTEM CLEAR command.

Response 2:

```
SYSTEM RESET
```

is issued by the SYSTEM RESET command.

Response 3:

```
STORE COMPLETE
```

is issued by the SYSTEM STORE STATUS command.

Response 4:

If you enter the command SYSTEM RESTART, no response is given; the virtual machine resumes execution at the address in the virtual PSW loaded from virtual storage location 0.

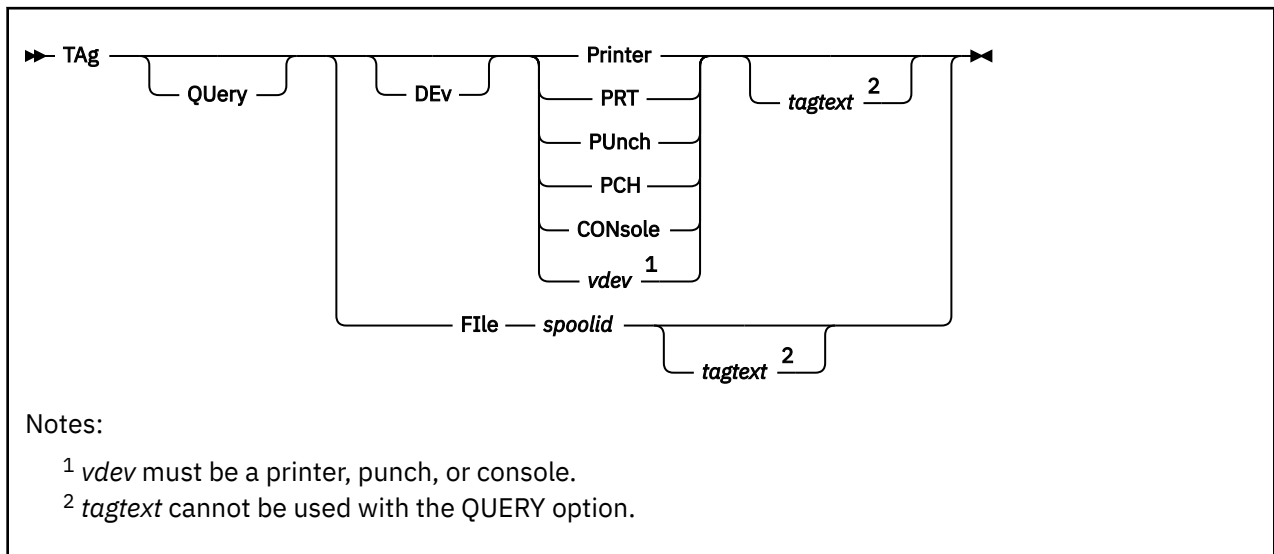
### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP410E CP ENTERED; PAGING ERROR
- HCP439E {User *userid*\*NSS}\*IMG}\*UCR}\*NLS} spool fileid limit exceeded
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP657E The virtual machine's page zero is unavailable because it is damaged.



- HCP799E IPL failed due to insufficient or faulty storage
- HCP830I The processor controller is responding slowly.
- HCP831I The processor controller is now responding.
- HCP832E The processor controller is disconnected due to a system error.
- HCP833I An IOCP has started a write sequence that will alter the I/O configuration data set *xx*.
- HCP834I The IOCP write processing cannot complete because the I/O configuration data set *xx* is invalid.
- HCP835I The IOCP write processing has completed; the I/O configuration data set *xx* is altered.
- HCP836I The IOCP read request cannot complete because *userid* already has a sequence in progress.
- HCP837I The IOCP read request cannot complete because *userid* already has a sequence in progress.
- HCP838E The request process cannot complete due to a processor controller failure.
- HCP839I The IOCP write open request cannot complete because data set *xx* is already open.
- HCP840I The IOCP write open request cannot complete because data set *xx* is already open.
- HCP842I The IOCP read request cannot complete because the data set is not open. The current open data set is *xx*.
- HCP843I The IOCP write request cannot complete because the data set is not open. The current open data set is *xx*
- HCP1455E CPU *nn* is not started because it is a secondary processor that is not operational.
- HCP1459E The virtual machine is placed in check-stop or system check-stop state due to a system malfunction with CPU *nn*.
- HCP6025E Operand {STATUS|STORE STATUS|RESTART} is invalid because CPU *nn* is in check-stop state.
- HCP6061I APPC wait canceled; virtual machine has been reset.
- HCP6086I Traceid *traceid1* no longer superseded by VMGROUP trace *traceid2*.
- HCP6525E Authorization request failed.

## TAG



## Authorization

Privilege Class: G

## Purpose

Use TAG to associate descriptive information with system spool files or spooled devices.

## Operands

### Q<sup>U</sup>ery

requests the display of the tag information associated with a particular virtual spooled device or spool file.

### D<sup>E</sup>v

specifies that a device is to be tagged or the tag information for a device is to be displayed.

### Printer

#### PRT

#### PUnch

#### PCH

#### CONsole

#### *vdev*

specifies a spooling device whose output is to be assigned the tag information. If you enter a generic device name, such as PRINTER or CONSOLE, all current virtual devices of that type are affected.

### File *spoolid*

indicates a closed spool file that you want to assign tag information. You can use this form of the TAG command only for spool files that are queued on your reader, printer, or punch queues.

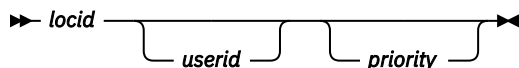
The operand *spoolid* is the spool file identifier CP assigns when the spool file is closed.

### *tagtext*

is the information you want to associate with the specified spool device or spool file. The contents and format of this data are flexible; they are the responsibility of the file originator and the end user.

## Usage Notes

1. If you enter the TAG command without *tagtext* information and without the QUERY option, the tag area associated with the device or file is set to all blanks.
2. Even though you have created a spool file, after you spool it to another virtual machine, the other user owns it as soon as the file is closed. If you want to query or change the tag data, you must first reclaim the file by issuing the CHANGE or TRANSFER command.  
**Note:** If an external security manager (ESM) is installed on your system, you may not be authorized to transfer spool files with the CHANGE or TRANSFER command.
3. You may not be authorized to enter the TAG command if an ESM is installed on your system. For additional information, contact your security administrator.
4. The buffer that contains tag information is 136 characters long, including blanks. However, because specific terminals may allow fewer than 136 characters, a user may not be able to type that many characters on the command line.
5. The tag information for a file is updated when the file is opened or closed. If a TAG command for a device is issued after a file has been opened, the tag information is updated when the file is closed. If a file is open during system shutdown or abend, the tag information in the file reflects the tag when the file was opened.
6. It should be noted that in an environment where RACF NODEID protection is implemented, TAG command data is assumed to be NODEID data when audited/controlled by RACF. Because of this, applications such as RSCS that regularly use the TAG command with data other than NODEID information can be adversely affected. In these cases RACF definitions should be adjusted to exclude these user ID's from audit/control.
7. The RSCS control program interprets the tag information as being addressing and control parameters. When you spool a file to the RSCS virtual machine to be transmitted to a remote station, code the *tagtext* operand as follows:



where:

### **locid**

is the node ID (one to eight alphanumeric characters) of the location to which the file is being transmitted. Your system programmer can give you the IDs of remote nodes accessible to your virtual machine.

### **userid**

is the user ID of the virtual machine (a one- to eight-character user identification) where a file is being transmitted. This operand is used by remote stations when they transmit files to a virtual machine in a remote spooling network and want the files sent to a particular virtual machine. You can ignore this operand if you are not specifying a priority. However, if you are specifying a priority, you must code some *userid* operand; it is ignored in a remote spooling network.

### **priority**

is the requested transmission priority, a decimal number between 0 and 99. The highest transmission priority is 0, next highest is 1, and so on. If you wish to specify this operand, you must also specify a *userid* operand.

8. Files that appear in PRINTED status when queried will not be selected for TAG processing.

## Responses

Response 1:

```
type vdev TAG:
tagtext
```

## TAG

displays the tag information associated with a virtual spooling device.

### **type**

is PRT, PUN, or CON.

### **vdev**

is the virtual device number.

### **tagtext**

is the tag information associated with the device. This tag information may be up to 136 characters long.

Response 2:

One of the following appears for the tag information associated with a specified spool file.

```
tagtext
```

```
(TAG BLANK)
```

### **tagtext**

is the tag information associated with the file. This tag information may be up to 136 characters long.

### **(TAG BLANK)**

is displayed when the tag is all blanks.

Response 3:

```
type {vdev } TAG NOT SET
```

displays the tag information associated with a device if:

- You have not specified tag information
- You have explicitly set the tag fields to blanks.

### **type**

is PRT, PUN, or CON.

### **vdev**

is the virtual device number.

## Messages

- HCP003E Invalid option - command contains extra option(s) starting with *option*
- HCP008E Invalid spoolid - *spoolid*
- HCP016E TAG invalid for specified file type
- HCP018E Spoolid missing
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP026E Operand missing or invalid
- HCP035E Device type missing or invalid
- HCP040E Device {*rdev*/*vdev*/*ldev*} does not exist
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP417E *rdev*/*vdev* is not an output spooling device.

# TERMINAL

---

## Purpose

Use the TERMINAL command to control various characteristics of your virtual console.

## Usage Notes

(General for TERMINAL)

1. You can specify more than one function in a single entry of the TERMINAL command. For example, `terminal chardel off linesize 80` specifies two terminal characteristics:
  - The character delete function CHARDEL is set OFF
  - The maximum line length for terminal output LINESIZE is set to 80 characters.
2. If an error occurs during processing of the command, all functions preceding the one with the error are in effect.
3. You cannot use any of the letters A through Z, the numbers 0 through 9, or the bytes X'0E' (shift out) or X'0F' (shift in) as a symbol (*char*) in association with any of the CHARDEL, LINEDEL, LINEND, TABCHAR, or ESCAPE operands.
4. Use the QUERY TERMINAL command to determine the values set for the TERMINAL command functions.
5. CP automatically masks your password on typewriter terminals; however, to maintain compatibility with previous releases, CP accepts the MASK operand of the TERMINAL command as valid.
6. The system operator may specify system defaults for the TERMINAL LINEND, LINEDEL, CHARDEL, TABCHAR, and ESCAPE characters with the Character Defaults statement in the system configuration file. See the [z/VM: CP Planning and Administration](#) manual.
7. The character displayed depends on the code page used by the terminal emulator. The following characters are from code page 037 United States:

Character	Meaning
# X'7B'	SYSLEND LINEND
¢ X'4A'	SYSLDEL LINEDEL
@ X'7C'	SYSLCEL CHARDEL
" X'7F'	SYSLESCP ESCAPE
X'6A'	SYSTAB TABCHAR
[ X'AD'	SYSALDEL LINEDEL

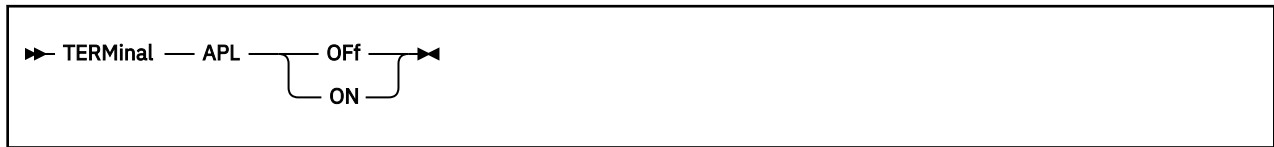
## Messages

- HCP002E Invalid operand - *operand*
- HCP002E Invalid operand - SHIFT IN (X'0F') (double-byte character set only)
- HCP002E Invalid operand - SHIFT OUT (X'0E') (double-byte character set only)
- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP262I The virtual console is not defined or is detached.

## TERMINAL

- HCP2104I DBCS output disabled by {APL|TEXT} ON because VTAM service machine controlling this terminal does not support DBCS and APL|TEXT output.
- HCP6005E Option *option* is not supported by z/VM.

## TERMINAL APL



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL APL to control APL character translation tables.

### Operands

#### OFF

#### ON

If you specify ON, CP uses the translation tables applicable to display stations equipped with the APL hardware feature. If you specify OFF, CP uses the normal terminal translation tables. Unless otherwise specified, APL OFF is in effect.

### Usage Notes

1. If you specify APL ON, TERMINAL TEXT is forced OFF.
2. TERM APL must be specified as OFF to allow a virtual machine to send mixed double-byte character set (DBCS) line mode output to its virtual console.
3. When you log on or reconnect, CP does not use the APL-character translation tables unless you specify the APL option on the LOGON command.
4. If the terminal is connected to VTAM through VSCS and the version of VSCS does not support DBCS, TERMINAL APL ON will cause DBCS output to be disabled. Error message HCPCFT2104I will be issued on the user's terminal screen.
5. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL ASCIITBL

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL ASCIITBL to select the translation table used by your TTY terminal.

### Operands

**VM1**

**VM2**

specifies the translation table to be used. VM1 specifies the use of the translation table based on the TTY ANSI X3.4 1977 Standard. VM2 specifies the use of the translation table based on the TTY ANSI X3.26 1980 Standard.

### Usage Notes

1. TERMINAL ASCIITBL may be issued only from a TTY device; otherwise an error message is issued.
2. ASCIITBL is set to VM1 when you first log on or reconnect.
3. For more information, see TERMINAL [“Usage Notes” on page 1917](#).



## TERMINAL ATTN

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL ATTN to control attention interrupts to ASCII terminal keyboards and printers.

### Operands

#### ON

turns on attention interrupt signaling. When an attention interrupt occurs, CP displays an exclamation point (!).

#### OFF

turns off attention interrupt signaling. When an attention interrupt occurs, CP suppresses the display of the exclamation point and carriage return.

**Note:** OFF is the default for display terminals. ON is the default for ASCII devices.

### Usage Notes

1. TERMINAL ATTN ON controls the signaling of an attention interrupt only on the keyboards and printers or ASCII terminals. It is not valid for display type terminals.
2. ATTN is set ON when you log on or reconnect to an ASCII terminal.
3. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL AUTO CR

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL AUTO CR to control whether a carriage return is performed following execution of the 3215 CCW X'01' (write without carriage return) to a graphic display operating in line mode.

### Operands

#### ON

#### OFF

controls carriage return. ON specifies that a subsequent line mode write-to-display starts on the line following the data just written. ON is the default.

OFF specifies that a subsequent line mode write-to-display start at the next column following the data just written.

### Usage Notes

1. AUTO CR is set ON when you first log on. Reconnecting does not affect the AUTO CR setting.
2. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL BREAKIN



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL BREAKIN to allow the user to control CP break-in while operating in full-screen mode.

### Operands

#### IMMed

specifies that CP can break in on any full-screen reads and full-screen writes that have an inactive keyboard, as well as when the user presses the BRKKEY.

#### GUESTctl

specifies that only high-priority warnings from CP and output initiated during DIAGNOSE X'08' processing (with the exception of the MESSAGE \* command) will break in. Other messages will break in only when the user presses the BRKKEY, or one of the previously mentioned tasks break in.

#### MINimal

specifies that high-priority warnings from CP and messages issued synchronously during DIAGNOSE X'08' processing will break in. Generally, synchronous message processing is required by those command processors that have the potential to generate large volumes of output and are interruptible in line mode by the user pressing the PA1 key. Examples are the DISPLAY storage commands, and many of the QUERY commands such as QUERY NAMES and QUERY DASD.

Other messages will break in only when the user presses the BRKKEY, or one of the previously mentioned tasks break in.

### Usage Notes

1. If you enter TERMINAL BREAKIN with the GUESTCTL option or the MINIMAL option from an SNA/CCS terminal, and the VTAM Service Machine (VSM) controlling that terminal only supports the IMMEDIATE option, then the command is not processed and you receive a message indicating a invalid device type error.
2. BREAKIN is set IMMEDIATE when you first logon. If you change it to GUESTCTL or MINIMAL and reconnect to a local display, then the setting is not changed. If you reconnect to a SNA/CCS display, and the VTAM Service Machine (VSM) controlling that display cannot support the IMMEDIATE option, then BREAKIN is reset to IMMEDIATE and a message is issued to inform you of the change.
3. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL BRKKEY



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL BRKKEY to define the key to control CP break-in for full-screen guests.

### Operands

#### **key**

must be one of the following: PA1, PA2, PF*nn*, or CLEAR. PF*nn* can be any PF key on your display station, from PF1 to PF24.

#### **NONE**

specifies that no BRKKEY can be defined.

### Usage Notes

1. The break key is in effect in line mode as well as in full-screen mode.
2. When you first logon, the default key to control break-in by CP for full-screen guests is PA1. However, if you intend to run an operating system in your virtual machine that requires the PA1 key for some other function, you can use this command to define an alternate break key.
3. After you enter this command, the new break key remains in effect for the duration of the session or until you reissue this command during the current session. If you use this command to define a new break key and then, at some point in the session, prefer to use the default break key again, the PA1 option lets you do so.
4. If you use this command to define a key other than PA1 as the break key, CP passes control of the PA1 key to your virtual machine.
5. When NONE is specified, the guest has complete control of the keyboard. If you are in full-screen you can only return to CP mode if the full-screen program provides a way for you or by typing #CP.
6. Entry to the protected application environment disables the BRKKEY. To reenale it within the protected application, enter the TERMINAL BRKKEY command. Upon exit from the protected application, CP restores the break key setting to what it was before entry into the same protected application. If the break key was changed while in the protected application, it remains unchanged when you exit the application.
7. For the BRKKEY option, only PA1 and NONE are supported for SNA/CCS terminals. If you enter TERMINAL BRKKEY with any other options from an SNA/CCS terminal, the command is not processed and you receive a message that indicates an invalid device type error.
8. If you change the break key setting while logged on to a local display, disconnect and reconnect to an SNA/CCS terminal. Your TERMINAL BRKKEY setting is affected in the following way during the reconnection processing:
  - If BRKKEY was previously set to NONE or PA1, it is not changed.
  - Otherwise, BRKKEY is set to PA1 and an informational message is issued to inform you of the change.

9. If a key other than PA1 is defined as the break key, then in line mode both the PA1 and this key will put the user in CP mode.
10. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL CHARDEL



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL CHARDEL to define the logical character-delete symbol for your virtual machine.

### Operands

**ON**

**OFF**

*char*

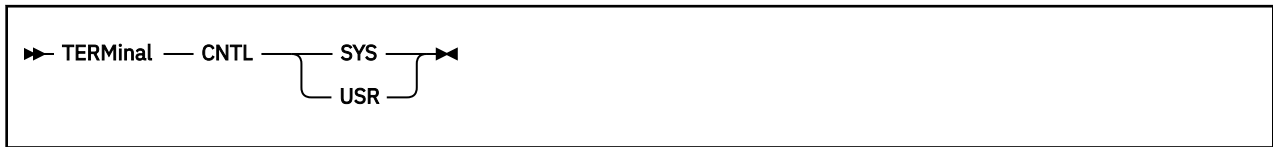
defines your logical character delete function. If you specify ON, the system default symbol becomes your logical character-delete symbol. If you specify OFF, no logical character-delete symbol is used for your virtual machine.

If you specify *char*, that character becomes your logical character-delete symbol. The character you select should be unique and not common to the data stream being entered.

### Usage Notes

1. When you first log on, CHARDEL is set to the system default symbol. The system default symbol is an at sign (@) unless changed in the system configuration file by the system operator. You can override this default by specifying a different setting in your directory entry. Reconnecting does not affect the CHARDEL setting.
2. For more information, see [“Usage Notes” on page 1917](#).

## TERMINAL CNTL



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL CNTL to allow CP or an application program to insert TTY line-control characters into your TTY terminal output.

### Operands

#### SYS

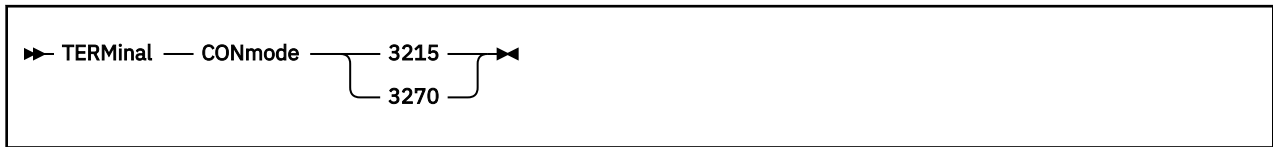
#### USR

determines whether CP or the application is to insert the TTY line-control characters. SYS specifies that CP inserts TTY line-control characters into output directed to a TTY terminal. USR specifies that the application program inserts TTY line-control characters into output directed to a TTY terminal.

### Usage Notes

1. TERMINAL CNTL may be issued from a TTY device only; otherwise, an error message is issued.
2. If you specify USR, both TERMINAL TYPE and TERMINAL PROMPT must be set to TTY. If TYPE and PROMPT are not TTY and you select USR, TYPE and PROMPT are forced to TTY. These options cannot be changed while CNTL is set to USR, or an error message is issued. The setting remains as TTY.
3. TERMINAL CNTL has no effect if an application issuing the CMS WRTERM macro does not specify either the EDIT=NO or EDIT=LONG parameters on the macro.
4. CNTL is set to SYS when you first logon. Reconnecting does not affect the CNTL setting.
5. For more information, see TERMINAL [“Usage Notes” on page 1917](#).

## TERMINAL CONMODE



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL CONMODE to control the console mode of your display.

### Operands

#### 3215

#### 3270

defines the type of virtual I/O support CP provides for your display. If you specify 3215, 3215 CCW processing occurs; if you specify 3270, 3270 CCW processing occurs.

### Usage Notes

1. In order to use this command correctly, you need to be aware of the factors that affect your console mode:
  - a. The CONSOLE statement in your system directory determines the initial console mode of your display.
  - b. You can enter the TERMINAL CONMODE command to change your console mode.
  - c. You can enter the DEFINE CONSOLE command to change your console mode.

In brief, there are two commands you can use to actively change your console mode: TERMINAL CONMODE and DEFINE CONSOLE.
2. In order to change your console mode setting with DEFINE CONSOLE, you must first detach your console with the DETACH command. For various reasons, you may not always want to detach your virtual console. TERMINAL CONMODE lets you change your console mode setting without detaching your console.
3. Certain programs require a console mode that may differ from your usual or default console mode. If you attempt to run such a program with the incorrect console mode, results are unpredictable. In most instances, the program terminates abnormally.

Some programs reset the console mode internally. For instance, CMS requires a console mode of 3215 and sets CONMODE to 3215 when it is IPLed. You may not be aware that the console mode has been changed for you automatically. Problems result if you then start a program that requires the alternate console mode setting.

Therefore, you should remain aware of the significance of the console mode any time that you start a program in your virtual machine.

4. When you define a virtual machine console with a console mode of 3270, the real terminal must be a graphics device. If you enter TERMINAL CONMODE 3270 from an SNA/CCS terminal controlled by a VTAM service machine that does not support CONMODE 3270, the command is not processed and you receive a message indicating a not-valid device-type error. CONMODE 3270 is supported by ACF/VTAM version 3, release 1.1 for VM/SP (VM/VTAM) and later releases.



5. CONMODE 3270 is incompatible with virtual integrated 3270 mode, which is enabled using SYS3270 ON..
6. Use QUERY TERMINAL to determine the console mode.
7. For more information, see TERMINAL [“Usage Notes” on page 1917](#).

## TERMINAL ESCAPE



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL ESCAPE to define the logical escape symbol for your virtual machine.

### Operands

**ON**

**OFF**

*char*

controls logical escape symbol selection. If you specify ON, the default symbol becomes the logical escape character for your virtual machine. If you specify OFF, no logical escape character is allowed for your virtual machine.

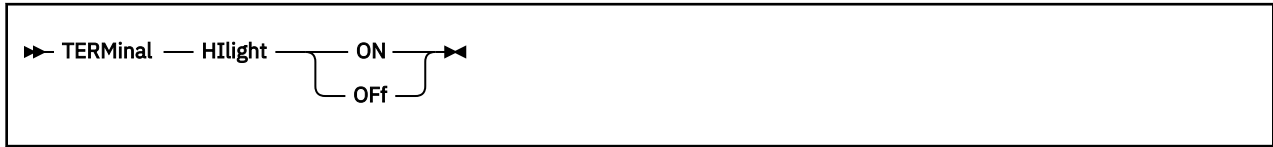
If you specify *char*, that character becomes the logical escape character. The character you select should be unique and not common to the data stream you are entering.

### Usage Notes

1. When you first log on, the logical escape symbol is set to the system default symbol. The system default is a double quotation mark (") unless changed in the system configuration file by the system operator. You can override this default by specifying a different setting in your directory entry. Reconnecting does not affect the ESCAPE setting.
2. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL HIGHLIGHT

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL HIGHLIGHT to highlight command redisplay on the virtual machine console.

### Operands

**ON**

**OFF**

controls command redisplay highlighting. When you specify OFF, user input is displayed at normal intensity. When you specify ON, user input is displayed at high intensity.

### Usage Notes

1. HIGHLIGHT is set to OFF when you first log on. Reconnecting does not affect the HIGHLIGHT setting.
2. The TERMINAL HIGHLIGHT command has no effect on the highlighting options specified on the SCREEN command.
3. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL HOLD



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL HOLD to control whether CP displays the HOLDING status when the terminal screen is full and highlighted output appears on the screen.

### Operands

#### ON

tells CP to display the HOLDING status when the terminal screen is full and highlighted output appears on the screen. To have CP display queued output, clear the screen manually by pressing the CLEAR or PA2 key. ON is the initial setting when you log on the system.

#### OFF

tells CP not to display the HOLDING status when the terminal screen is full and highlighted output appears on the screen. You do not have to clear the screen manually. CP still switches the screen status to HOLDING if you enter a null command when the screen is in the MORE . . . state.

### Usage Notes

1. The TERMINAL HOLD command will give you an error message, HCP006E, if you enter it on a TTY device.
2. If you modify the HOLD value to something other than the default value of ON and reconnect to a SNA terminal controlled by a VTAM Service Machine that does not support the TERMINAL HOLD command, message HCP2556I is issued to inform you that the HOLD value has changed back to ON.
3. If you specify a HOLD value of OFF on a SNA terminal controlled by a VTAM Service Machine that does not support the TERMINAL HOLD command, message HCP006E is issued.
4. See [“TERMINAL MORE” on page 1937](#) for information on how to change the number of seconds that elapse between the time when CP:
  - issues the MORE . . . state and sounds the terminal alarm
  - sounds the alarm and clears the screen.

### Messages

- HCP002E Invalid operand - *operand*
- HCP006E Invalid device type - *vdev*
- HCP026E Operand missing or invalid

## TERMINAL LINEDEL



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL LINEDEL to define the logical line-delete symbol for your virtual machine.

### Operands

**ON**

**OFF**

***char***

defines the logical line-delete function. If you specify ON, the system default symbol becomes the logical line-delete symbol for your virtual machine. If you specify OFF, no logical line-delete symbol is allowed for your virtual machine.

If you specify *char*, that character becomes the logical line-delete symbol. The character you select should be unique and not common to the data stream you are entering.

**Note:** For ASCII devices the default symbol is a left bracket ([).

### Usage Notes

1. When you first log on, LINEDEL is set to the system default symbol. The system default symbol is a cent sign (¢) unless changed in the system configuration file by the system operator. You can override this default by specifying a different setting in your directory.
2. If your LINEDEL is the cent sign (¢) and you reconnect to an ASCII device, LINEDEL is changed to the left bracket ([). If your LINEDEL is the left bracket ([) because you were on an ASCII device earlier, LINEDEL is changed back to the cent sign (¢).
3. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL LINEND



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL LINEND to define the logical line-end symbol for your virtual machine.

### Operands

**ON**

**OFF**

***char***

controls logical line-end symbol definition. If you specify ON, the system default symbol becomes the logical-line end symbol for your virtual machine. If you specify OFF, no logical line end symbol is allowed for your virtual machine.

If you specify *char*, that character becomes the logical line-end character. The character you select should be unique and not common to the data stream you are entering.

### Usage Notes

1. When you first log on, LINEND is set to the system default symbol. The system default symbol is a pound sign (#) unless changed in the system configuration file by the system operator. You can override the default by specifying a different setting in your directory entry. Reconnecting does not affect the LINEND setting.
2. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL LINESIZE



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL LINESIZE to control display screen line length.

### Operands

*nnn*

OFF

controls allowable line length. The value you specify for *nnn* can be any number from one through 255. OFF specifies that the output line length is not to be adjusted by CP; in other words, CP does not split a long output line into two or more shorter ones.

### Usage Notes

1. If TERMINAL APL ON is specified, CP defines an output length of 1760 and assumes that the APL system has inserted the appropriate carriage-control characters into the output data.
2. If LINESIZE is set too small to display one byte of data per line, CP uses the smallest value for LINESIZE that it can in order to display one byte of data per line. For instance, if LINESIZE=2 but the message to be displayed consists of mixed DBCS data, the smallest value needed to display one byte of DBCS data is 4, so this value will display that message.
3. LINESIZE is set to the character width of the terminal when you log on or reconnect.
4. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL MODE

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL MODE to control the display station attention environment.

### Operands

**CP**

**VM**

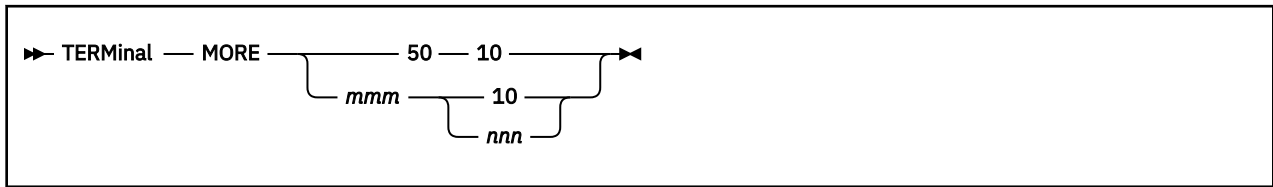
controls the attention environment. If you specify CP, one or more attentions force your virtual machine into the CP environment. If you specify VM, one or more attentions are reflected to your virtual machine. VM is the default for all users except the primary system operator.

### Usage Notes

1. This command has meaning only in line mode.
2. For more information, see TERMINAL [“Usage Notes” on page 1917](#).



## TERMINAL MORE



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL MORE to change the number of seconds that elapse between the time when CP issues the MORE . . . state and sounds the terminal alarm and between the time when CP sounds the alarm and clears the screen.

### Operands

#### *mmm*

tells CP the number of seconds that should pass after the MORE . . . status is displayed before CP sounds the terminal alarm. *mmm* must be a number between 0 and 255. When you first log on the system, CP sets *mmm* to 50 seconds.

#### *nnn*

tells CP the number of seconds that should pass after CP sounds the terminal alarm before it clears the screen. *nnn* must be a number between 0 and 255. When you first log on the system, CP sets *nnn* to 10 seconds.

Initial Setting:

50 10

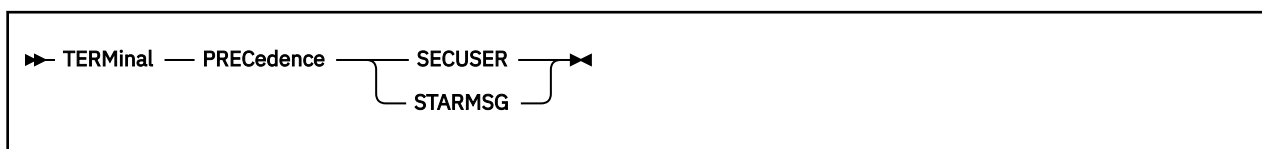
### Usage Notes

1. The TERMINAL MORE command will give you an error message, HCP006E, if you enter it on a TTY device.
2. If you specify MORE values other than 50 10 on a SNA terminal controlled by a VTAM Service Machine that does not support the TERMINAL MORE command, message HCP006E is issued.
3. If you modify the MORE values to something other than the default values of 50 10 and reconnect to a SNA terminal controlled by a VTAM Service Machine that does not support the TERMINAL MORE command, message HCP2555I is issued to inform you that the MORE values have changed back to 50 10.
4. See “TERMINAL HOLD” on page 1932 for information on how to prevent CP from presenting a HOLDING status when the terminal screen fills up and highlighted messages appear on the screen.

### Messages

- HCP002E Invalid operand - *operand*
- HCP006E Invalid device type - *vdev*
- HCP026E Operand missing or invalid

## TERMINAL PRECEDENCE



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL PRECEDENCE to determine how the issuer's virtual console output is delivered when it is eligible for routing to both a secondary user and the local virtual machine through IUCV. Output is eligible for routing to the secondary user when

1. A secondary user is set.
2. The secondary user is logged on.
3. Either the secondary user is connected to a console or has a connection to \*MSG or \*MSGALL.
4. If mandatory access controls are active, the SECLABEL of the secondary user either is equivalent to or dominates that of the primary user.

Output is eligible for routing to the local virtual machine through IUCV when

1. The user has a connection to \*MSGALL, or
2. The user has a connection to \*MSG and the control governing the routing of the class of output being written (SET MSG, SET WNG, SET EMSG, SET IMMSG, SET SMSG, SET CPMCONIO, or SET VMCONIO) is set to IUCV.

### Operands

#### SECuser

#### STARmsg

SECUSER means that output eligible for delivery to both a secondary user and to the local virtual machine via IUCV is delivered to the secondary user. STARMMSG means that such output is delivered to the virtual machine via IUCV.

### Usage Notes

1. If the virtual console output is not eligible for routing as described above then this setting has no effect on how it is delivered.
2. The default setting at LOGON is SECUSER.

## TERMINAL PROMPT

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL PROMPT to control the prompting sequence in a TTY device.

### Operands

#### VM

#### TTY

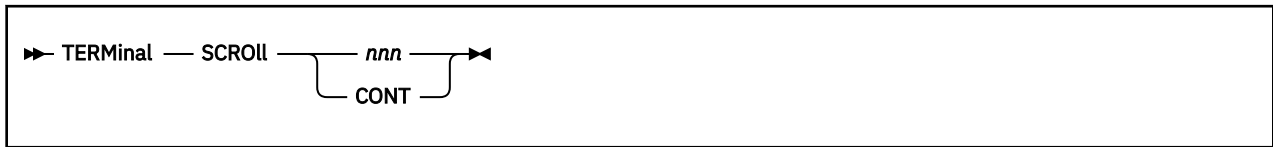
controls cursor position. The VM option causes the cursor to be positioned in column 1 of the input line for a read operation. The TTY option enables the normal TTY prompt sequence of a period in column 1 and the cursor in column 2.

### Usage Notes

1. TERMINAL PROMPT can be issued from a TTY device only; otherwise, an error message is issued.
2. If a user disconnects from the system and then logs on again, the PROMPT options are reset to the system defaults. PROMPT is set to TTY when you first logon or reconnect.
3. For more information, see TERMINAL [“Usage Notes” on page 1917](#).

## TERMINAL SCROLL

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL SCROLL to control the number of display screen lines scrolled.

### Operands

*nnn*

**CONT**

controls screen lines scrolled. The option *nnn* specifies the number of lines scrolled up before scrolling stops with an indication of more lines to follow. Pressing the NEW LINE key (character mode) or the SEND key (block mode) causes another cycle of *nnn* lines to be scrolled. For normal frame by frame scrolling, make *nnn* equal to the number of data lines on the screen. The operand CONT specifies continuous scrolling to the end of the output.

### Usage Notes

1. TERMINAL SCROLL can be issued from a TTY device only; otherwise, an error message is issued.
2. SCROLL is set to CONT when you first logon or reconnect.
3. For more information, see TERMINAL [“Usage Notes” on page 1917](#).

## TERMINAL SYS3270



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL SYS3270 to control whether the virtual machine console (3215) emulates an integrated 3270 device.

### Operands

#### **SYS3270**

#### **SYSGRAF**

SYSGRAF is a synonym for SYS3270.

#### **ON**

#### **OFF**

determines if CP provides virtual integrated 3270 support for the console device.

### Usage Notes

1. In order to use this command correctly, you need to be aware of the factors that affect your console mode:
  - a. You can enter the TERMINAL SYS3270 command to control whether your virtual machine console emulates an integrated 3270 device.
  - b. From the virtual machine's perspective, issuing SYS3270 ON has the same effect as opening the Integrated 3270 window on the Hardware Management Console.
  - c. From the virtual machine's perspective, issuing SYS3270 OFF has the same effect as closing the Integrated 3270 window on the Hardware Management Console.
  - d. When you specify SYS3270 ON, the real terminal must be a 3270 device.
  - e. SYS3270 ON is incompatible with 3270 console mode, which is enabled using CONMODE 3270.
  - f. The guest operating system must not use the virtual console as a conventional I/O device, despite the fact that it has a virtual device number and subchannel identifier. The real integrated 3270 device does not have these characteristics, so the guest must behave as though they do not exist. For example, the device may need to be logically varied offline before its mode is changed to SYS3270 ON. Otherwise, device behavior may be unpredictable.
2. Use QUERY TERMINAL to determine if the virtual console is emulating a 3270 device. The response section of the query will display SYS3270 ON or SYS3270 OFF.
3. For more information, see TERMINAL [“Usage Notes” on page 1917](#).

## TERMINAL TABCHAR



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL TABCHAR to define logical tab characters for a display device.

**Note:** QUERY TERMINAL displays the current logical tab character.

### Operands

**ON**

**OFF**

*char*

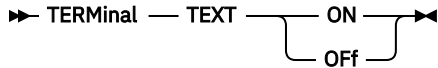
defines logical tab characters. ON sets the logical tab character to the system default symbol. OFF resets the logical tab character to undefined. The operand *char* defines the logical tab character. It may be any valid specific graphic character—that is, a character other than A through Z or 0 through 9.

### Usage Notes

1. When a not valid character is specified, an error message is issued to the display, and the logical tab character remains unchanged.
2. When you first log on, TABCHAR is set to the system default symbol. The system default symbol is a vertical broken bar (|) unless changed in the system configuration file by the system operator. Reconnecting does not affect the TABCHAR setting.
3. For more information, see TERMINAL [“Usage Notes” on page 1917](#).

## TERMINAL TEXT

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL TEXT to control the use of text translation tables.

### Operands

#### ON

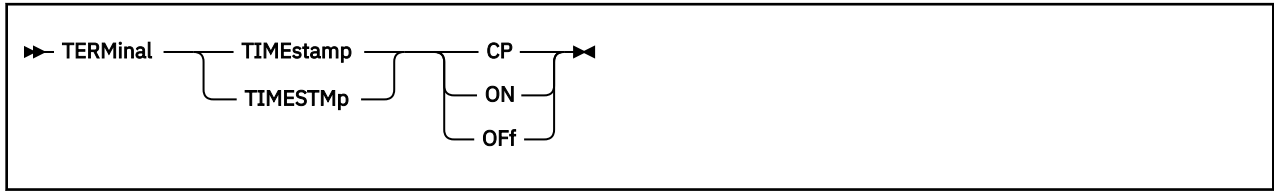
#### OFF

If you specify ON, CP uses the translation tables applicable to display terminals equipped with the text hardware features. If you specify OFF, CP uses the normal translation tables. Unless otherwise specified, TEXT OFF is in effect.

### Usage Notes

1. If TEXT is set ON, a previous specification of TERMINAL ON is negated.
2. TEXT must be set OFF to allow the virtual machine to send mixed double-byte character set (DBCS) line mode output to its virtual console.
3. If the terminal is connected to VTAM through VSCS and the version of VSCS does not support DBCS, TERMINAL TEXT ON will cause DBCS output to be disabled. Error message HCP2104I will be issued on the user's terminal screen.
4. For more information, see TERMINAL [“Usage Notes”](#) on page 1917.

## TERMINAL TIMESTAMP



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL TIMEStamp/TIMESTMp to control whether messages and command responses on the virtual machine console are prefixed with the local time of day.

### Operands

**CP**

**ON**

**OFF**

If you specify CP, control program messages and command responses on the console are prefixed with the local time of day. ON causes all virtual machine and CP messages to be identified with time stamps. If you specify OFF, messages are displayed with no prefix.

### Usage Notes

1. Time stamps are not produced for some CP command responses (for example, to QUERY READER ALL) to preserve readability.
2. Messages sent to a secondary user for a primary user with TIMEStamp CP are not prefixed with a time stamp, for compatibility with earlier releases of VM. The time stamp prefix generation that occurs for a secondary user depends on the TIMEStamp settings of both the primary and secondary users. The sources of time stamps are summarized in the table below.

*Table 45. Time Stamp Prefix Sources*

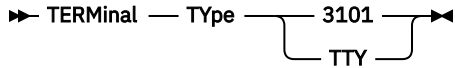
Secondary User TIMEStamp Setting	Primary user TIMEStamp Setting		
	CP	ON	OFF
CP	None	Primary user	None
ON	Secondary user	Both users	Secondary user
OFF	None	Primary user	None

3. The system default is TIMEStamp OFF except for the primary system operator whose TIMEStamp setting is CP.



## TERMINAL TYPE

---



### Authorization

Privilege Class: G

### Purpose

Use TERMINAL TYPE to define a device as a 3101 or a TTY.

### Operands

#### 3101

#### TTY

defines the device.

- When TTY is specified, the terminal is regarded as a typewriter terminal.
- When 3101 is specified, the terminal is regarded as a display screen and keyboard ASCII device. The 3101 operand:
  1. Allows the use of program function keys 1 through 24
  2. Causes output to begin at column 1 of line 24 on the display
  3. Allows editing of previously entered information without requiring that the entire text be rekeyed.

### Usage Notes

1. TERMINAL TYPE can be issued from a TTY device only; otherwise, an error message is issued.
2. The device is defined as a TTY when you first log on or reconnect.
3. For more information, see TERMINAL [“Usage Notes” on page 1917](#).

## TRACE

---

### Authorization

Privilege Class: G

### Purpose

Use the TRACE command to monitor events that occur in your virtual machine. z/VM lets you trace a number of events, including:

- Instruction execution
- Storage alteration
- Register alteration
- I/O activity.

Each traced event results in a *trace entry*, a command response that you can have sent to your virtual console, to a virtual printer, or to both. The trace entry is made up of significant information about the event. You can use trace entries to analyze the operation of your virtual machine and to debug problems.

Note that in this manual the TRACE documentation is primarily for reference.

For a full explanation of *how* to use the trace facility, along with examples, see [Tracing Programs in Your Virtual Machine in z/VM: Virtual Machine Operation](#).

### Usage Notes

(General for TRACE)

1. All of the primary TRACE operands establish *trace traps*. A trace trap is a programming function that, as its name implies, captures information about an event in your virtual machine.

When you establish a trap, you can assign it any unique 1-character to 4-character name (with the exception of the name ALL). If you do not assign a name, the system assigns the trap a unique number from 1 to 9999. You use this unique identifier (name or number) to alter a trap with the TRACE TRAP command or to delete a trap with the TRACE DELETE command.

The options (unique and common) available with each primary operand and with the TRAP operand allow you to specify exactly how you want the trap to process. For example, you may want the trap to function only when an event occurs within a particular address range. In this case, you specify the common option FROM with the address range you choose. At the same time, you may want to trace events only when your virtual machine is running in supervisor state. In this case, you also specify the common option SUPERVISOR.

2. All trace traps must be contained in units of control called sets. A “trace set” is a collection of one or more traps that run together.

You create a set using the TRACE GOTO command. When you create the set, you assign it a unique 1-character to 8-character name. If you do not assign the set a name, the system assigns a unique name for you.

If you do not create a set before you establish a trap, z/VM creates a set for you named INITIAL.

Only one set may be active at any given time.

The following commands exist for manipulating sets:

- TRACE GOTO
- TRACE CALL
- TRACE RETURN
- TRACE CLEAR

- TRACE END
  - TRACE APPEND.
3. You can define up to 255 trace sets and trace traps in any combination during a single trace session. (For example, you can create two trace sets with no more than 253 traps between them, five trace sets with no more than 250 traps between them, or 20 trace sets with no more than 235 traps between them).
  4. If you have RUN set ON, it will be turned off when a trace trap is established in a non-suspended trace set.
  5. For information on how to display your current trace settings, see the QUERY TRACE command. The QUERY TRACE command can be especially useful if you need to know the identifiers the system has assigned to your traps or sets. The command QUERY TRACE ALL displays the attributes of all trace traps within all trace sets.
  6. The IPL command for z/VM suspends all user tracing through the TRACE command during the IPL process. Tracing is then automatically reset as it was and continues normally.
  7. DATOFF is a synonym for NODAT. DATOFF is allowed whenever NODAT is allowed.
  8. Depending on the model, TRACE events for general register alteration may or may not be recognized for vector instructions.
  9. Self-modifying instructions or instructions that change the contents of registers used to compute operand addresses cause erroneous processing of trace output. For this reason, question marks (?) are displayed in the address field instead of erroneous output.
  10. If your virtual machine initiates full-screen I/O operations to your virtual console while CP is displaying trace entries there, the full-screen I/O operations may not complete successfully. This is most likely to occur if TERMINAL CONMODE has been set to 3270.
  11. The TRACE command will fail if issued when VMRELOCATE is in progress.
  12. When z/VM does not provide full trace support for an instruction, the command response will not include operand addresses. See TRACE RESPONSE for more details

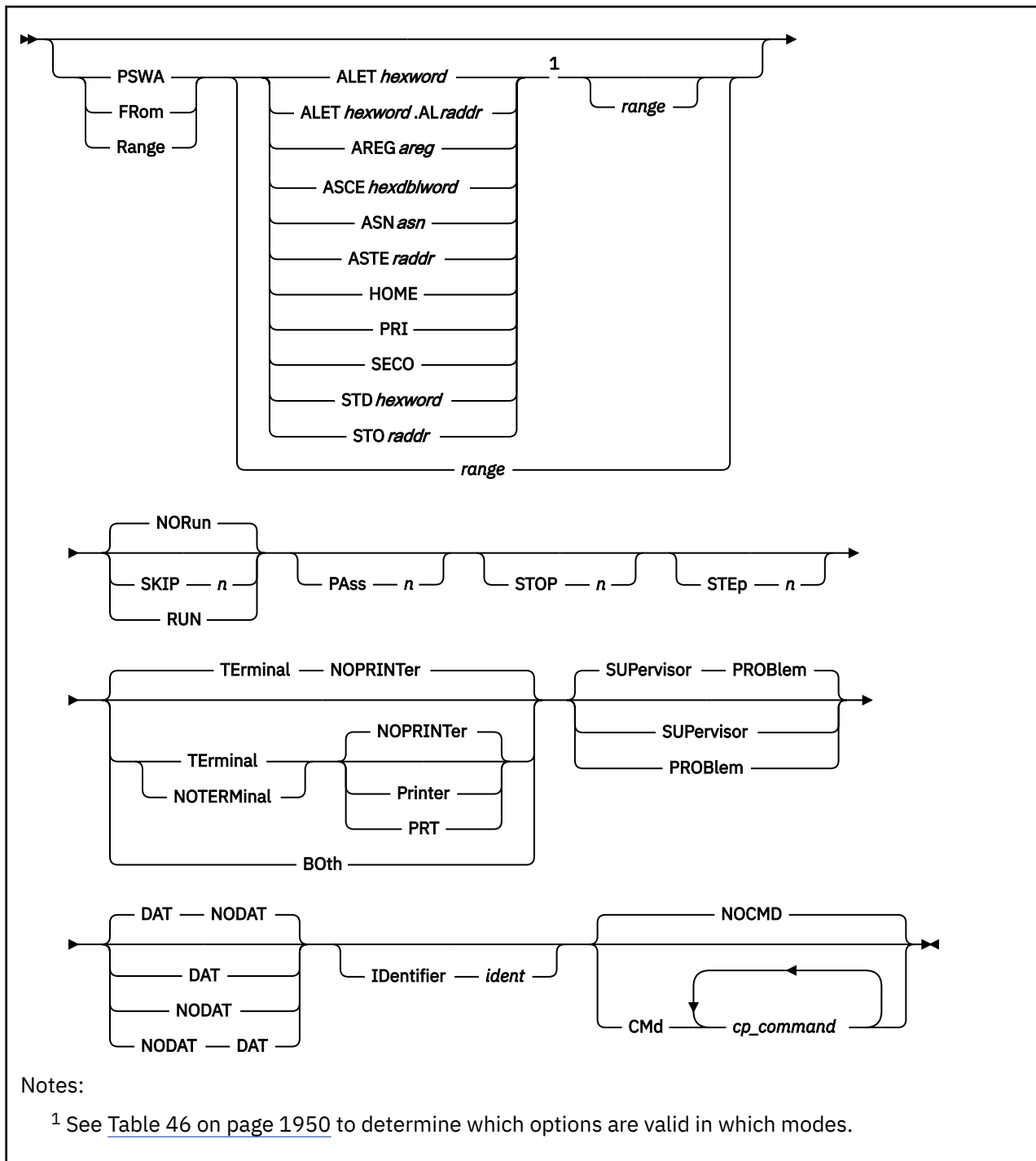
## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option|command contains extra option(s) starting with option}*
- HCP009E Invalid range - *range*
- HCP010E Invalid register - *register*
- HCP013E Conflicting option - *option*
- HCP026E Operand missing or invalid
- HCP033E Hexloc missing or invalid
- HCP167E The mask is invalid for the specified data range.
- HCP180E Trace not in effect
- HCP1001E An operand is missing for - *option*
- HCP1005E An invalid character was encountered.
- HCP1025E A recursive trace call cannot be done.
- HCP1026E The trace return cannot be done because there is no matching trace call.
- HCP1027I An active trace set has turned RUN off.
- HCP1032E A syntax error was encountered.
- HCP1035E A trace set cannot be appended to itself.
- HCP1036E *{traceset|The current trace set}* is an empty trace set.
- HCP1037E The command cannot be processed because the maximum number of TRACE traps is already defined.

## TRACE

- HCP1040E TRACE suspended - Incompatible with virtual machine's use of PER instruction-fetching-nullification facility.
- HCP2106I TRACE caused a guest transaction at *tbegin\_address* to be aborted at *abort\_address*
- HCP2107E The virtual machine's use of PER Event Suppression conflicts with TRACE *txsusp*.
- HCP6150E An option has been specified that is not valid in {ESA/390|ESA/XC|z/Architecture|z/XC} mode.
- HCP6151E Storage {operand} is not addressable because of { a protection |an addressing | a segment translation | a page translation | a translation specification | an ASN translation | an AFX translation | an ASX translation | an ALET translation | an ALEN translation | an ALE sequence | an ASTE validity | an ASTE sequence } exception.
- HCP6156E Access register | ALET |ASIT | Address Space Identifier (spaceid) does not designate an address space to which you have the appropriate access.
- HCP6577E Invalid ALET - *alet*
- HCP6578E Invalid {access register|ASIT|address space identifier} - *option*

## TRACE: Options



### Purpose

If you specify any of these options without a primary trace operand, they are applied to all events in the current trace set. If you specify the TRAP operand, these operands are applied to a single event.

### Options

[Table 46 on page 1950](#) shows which TRACE options are valid in which modes.

Option	ESA/390	z/Architecture	ESA/XC, z/XC
ALET	x	x	x
AREG	x	x	x
ASCE	x	x	
ASN	x	x	
ASTE	x	x	
HOME	x	x	
PRI	x	x	x
SECO	x	x	
STD	x		
STO	x		

**PSWA range****FRom range****Range range**

specifying the program status word address (PSWA) restricts instruction tracing to a given address range. It permits a trap to be taken if the PSWA (commonly known as the instruction counter) points to an instruction in the specified range. The range may be a single address, a pair of addresses separated by a hyphen (-), or an address followed by a period (.) followed by a byte count. An instruction is considered to be within the range if the first byte is within the range.

The *FROM* option is not supported for the TRACE AIF command.

The instruction addresses specified for the range cannot exceed FFFFFFFF FFFFFFFF.

FROM and RANGE are synonyms for PSWA used in those subcommands where a potential for confusion exists. For example, TRACE BRANCH allows two instruction ranges to be specified, a FROM range and an INTO range. FROM and RANGE are allowed wherever PSWA is allowed.

For executed instructions (that is, instructions that are issued by means of the assembler language EXECUTE instruction), the instruction is within the range if either:

- The first byte of the EXECUTE instruction is within the range
- The first byte of the target instruction is within the range.

For external interrupts, the PSW address at the time of the interrupt is inspected to see if it is within the range.

When a program event recording (PER) event is reflected to your virtual machine, the PER event address is checked.

For any other program exception, the old PSW instruction address is inspected if a nullification exception occurs; for all others, the old PSW instruction address is adjusted by the instruction length.

**ALETheyword**

indicates that access register translation be used to locate the address space to be traced. The access list entry token (ALET) is used in conjunction with the guest access register translation tables to locate the STO of the address space to be traced.

**ALETheyword.ALraddr**

restricts tracing to events in an address space using the specified access list entry token (ALET) and access list (AL). The ALET and AL are used in conjunction with the guest access register translation tables to locate the STO of the address space to be traced.

**AREGareg**

indicates that access register translation be used to locate the address space to be traced. The access register (*areg*) is specified as one hexadecimal or two decimal digits. The access list entry token (ALET) within the access register is used in conjunction with the guest access register translation tables to locate the STO of the address space to be traced.

**ASCEhexdblword**

indicates that an address space control element (ASCE) be used to locate the address space to be traced.

**ASNasn**

indicates that an address space number (*asn*) be used to locate the address space to be traced. The specified ASN may be one to four hexadecimal digits. The ASN is used in conjunction with guest ASN translation tables to locate the STO of the address space to be traced.

**ASTERaddr**

indicates that an ASN second table entry (ASTE) be used to locate the address space to be traced. The operand *raddr* is the guest real address of the ASTE that contains the segment table origin (STO) of the address space to be traced.

**HOME**

restricts tracing to events in the guest's current home address space as pointed to by the guest control register 13.

**PRI**

restricts tracing to events in the guest's current primary address space as pointed to by the guest control register 1.

**SECO**

restricts tracing to events in the guest's current secondary address space as pointed to by the guest control register 7.

**STDhexword**

indicates that a segment table designation (STD) be used to locate the address space to be traced. The STD contains the segment table origin (STO) of the address space to be traced.

The STD option is not valid in z/Architecture mode.

**STOraddr**

indicates that a segment table origin (STO) be used to locate the address space to be traced. The operand *raddr* is the segment table origin of the address space to be traced and is a guest real address.

The STO option is not valid in z/Architecture mode.

**SKIP n**

requests that the associated event be completely ignored *n* times. When used with the NOSIM option, however, it specifies the number of times that an instruction is simulated normally.

**RUN**

specifies that virtual machine execution is not to pause when the specified event occurs. RUN is assumed if TERMINAL is not specified or assumed. RUN is incompatible with NORUN, STEP, and STOP.

**NORun**

specifies that virtual machine execution is to pause when the specified event occurs. NORUN is assumed if TERMINAL is specified or assumed. NORUN is incompatible with RUN, STEP, and STOP.

**PAss n**

is a repeating SKIP specification. If you do not specify a SKIP option, PASS takes effect immediately. If you do specify a SKIP count, PASS takes effect after the SKIP count expires and the specified event has been recognized.

For example:

- PASS 2 displays every third event
- PASS 8 displays every ninth event
- SKIP 2 PASS 8 displays the third event, the twelfth event, and every ninth event thereafter.

**STOP *n***

requests a number of displays for the command. Your virtual machine pauses after the specified number of displays. Unless a STEP count is specified, a pause occurs for each subsequent display.

**STEp *n***

is a repeating STOP specification. If you do not specify a STOP option, STEP takes effect immediately. If you do specify a STOP count, STEP takes effect after the first pause.

For example:

- STEP 3 pauses every third event
- STEP 9 pauses every ninth event
- STOP 3 STEP 9 pauses after the third and twelfth displays and for every ninth display thereafter.

**TErMinal**

specifies that output for the current command be sent to the virtual console. This is the default setting for newly created tracing specifications with neither terminal nor printer output specifications. TERMINAL and NOTERMINAL are incompatible options.

**NOTERMinAl**

specifies that output for the current command is not to be sent to the virtual console. TERMINAL and NOTERMINAL are incompatible options.

**BOth**

specifies that the output for the current command be sent to the virtual console and to the virtual printer.

**Printer**

specifies that output for the current command be sent to the virtual printer having the lowest virtual device number. PRINTER may be abbreviated to PRT; PRINTER and NOPRINTER are incompatible options.

**NOPRINTEr**

specifies that output for the current command is not to be sent to the virtual printer. NOPRINTER may be abbreviated to NOPRT; PRINTER and NOPRINTER are incompatible options.

**SUPervisor**

requests that the specified tracing be restricted to events occurring while the machine is in supervisor state. If specified in addition to PROBLEM, or if neither SUPERVISOR nor PROBLEM is specified, both problem state and supervisor state events are traced.

**PROBLEM**

requests that the specified tracing be restricted to events occurring while the machine is in problem state. If specified in addition to SUPERVISOR, or if neither SUPERVISOR nor PROBLEM is specified, both problem state and supervisor state events are traced.

**DAT**

requests that the specified tracing be restricted to events occurring while the virtual machine has virtual dynamic address translation enabled. If you specify DAT in addition to NODAT, or if you specify neither DAT nor NODAT, events occurring in both DAT and non-DAT mode are traced.

**NODAT**

requests that the specified tracing be restricted to events occurring while the virtual machine has virtual dynamic address translation disabled. If you specify NODAT in addition to DAT, or if you specify neither DAT nor NODAT, events occurring in both DAT and non-DAT mode are traced.

**IDentifier *ident***

specifies an identifier for a trace trap describing the event. It must be one to four characters long. If a trap already exists with the given identifier, it is replaced by the trap for the newly defined event. If you do not specify an ID, a unique number from 1 to 9999 is assigned as the identifier, with leading zeros removed. You cannot use the word ALL as an identifier. You may specify the IDentifier option only when defining a trace trap, not when modifying one with TRACE TRAP or the TRACE command without a primary trace operand.

**CMd *cp\_command***

specifies a CP command or set of CP commands to be processed when the specified event occurs.



Multiple commands must be separated by line-end characters. Actual line-end characters end the trace command. For example, if the pound sign (#) is your logical line-end character, and if the double quotation mark (") is your logical escape character, **trace svc cmd d 20#d 30** issues a TRACE SVC command that displays location 20, and then issues DISPLAY to display location 30.

The command **trace svc cmd d 20"d 30** issues one TRACE command; it displays locations 20 and 30 whenever an SVC instruction is executed.

You can use this option with TRACE GOTO, TRACE CALL, and TRACE RETURN to allow conditional and unconditional branching and subroutine calls between trace environments.

All characters between the CMD option and the end of the TRACE command line are considered part of the CMD field; thus, it must be the last option on the line. No command syntax verification is made on the set of CP commands to be processed until the specified event occurs. Terminal output from the set of commands is not affected by the TERM or NOTERM specification of the trace event.

### NOCMD

specifies that no CP command be run when the specified event occurs. Use NOCMD to delete CMD fields from existing trace events.

### Usage Notes

1. Guest control register values and translation table contents define the address space structure of the virtual machine. Program execution (and the STORE command) can change that structure over time. The TRACE command operands that specify a guest address space to be traced by a trace trap use the current guest control register and translation table values to determine the segment table origin (STO) of the address space to be traced (at the time the TRACE command is issued). The STO or ASCE is contained in the trace trap and can be examined by use of the QUERY TRACE command—that is, you specify an address space based on the state of your virtual machine at the time the TRACE command is issued.
2. A trace trap generated with a specific address space specification (STO or ASCE) always identifies that address space using the STO or ASCE generated from the control registers, the translation tables, or both. If your program execution changes the control register values and translation table contents to represent a different address space structure, there is *no* change to the trace traps. The trace traps remain in effect with the address space specified at the time of the TRACE command execution.
3. If a guest specifies an address space option from which an ASN can be derived, then trace will trap events in all subspaces which are members of the group identified by the ASN whenever possible.

An ASN can be derived from the following address space options.

- PRI
  - SECO
  - ASN
  - AREG0
  - AREGx, where the ALET in ARx is zero or one
  - ALETn, where n is zero or one
4. For more information, see TRACE [“Usage Notes” on page 1946](#).

## TRACE: Command Responses

---

### Purpose

In this section are illustrated the command responses for each of the TRACE command operands. These response descriptions use the following symbols:

**aaaaaaaa**

**aaaaaaaaaaaaaaaaaaaa**

**aaaaaaaa\_aaaaaaaa**

A storage address, followed by a single quotation mark (') if a primary address space storage address or a double quotation mark (") if a home address space storage address.

For external and I/O interruptions, it is the instruction address at the point of interruption, taken from the corresponding old PSW.

For program interruptions, it is the address of the instruction that encountered the program interruption. More specifically:

- For a PER event (either alone or accompanied by some other program interruption condition), it is the PER event address.
- For a nullifying program interruption not accompanied by a PER event, it is the instruction address in the program old PSW.
- For a suppressing, terminating, or completing program interruption not accompanied by a PER event, it is the instruction address in the program old PSW, minus the instruction length code.<sup>1</sup>

When a program interruption occurs during transactional-execution mode, two addresses are generally reported. First, the instruction at the point of the transaction abort is shown. Then, on the program-interruption (PROG) line, the address at left shows the address of the TBEGIN or TBEGINC instruction of the outermost aborted transaction. In some cases, the location of the abort is not known; the instruction display line is then omitted, and the PROG line shows the instruction address from the program old PSW. This is either the address after the TBEGIN or the address of the TBEGINC.

**tttttttt**

**tttttttt\_tttttttt**

**tttttttttttttttt**

A transfer address, followed by a single quotation mark (') if DAT.<sup>2</sup>

**xxxxxxxx**

An instruction, CSW status, and other such items.

**gggggggg**

A second-level channel command word, CSW status, and other guest output.

**hhhhhhhh**

A first-level channel command word, CSW status, and other such host output.

**oooo**

A first-level CCW offset.

**address1**

The address of operand 1 (for SI and SS formats only).<sup>2</sup>

**address2**

The address of operand 2 (for RX, RS, S, and SS formats only).<sup>2</sup>

<sup>1</sup> The number of hexadecimal digits displayed depends on the virtual machine mode: eight for ESA/390-mode and ESA/XC-mode machines; 16 for z/Architecture-mode and z/XC-mode machines. If SET UNDERSCORE is set on, the 16 digits are separated in the middle with an underscore for easier readability.

<sup>2</sup> The number of hexadecimal digits displayed depends on the guest addressing mode: eight for 24-bit or 31-bit addressing mode; 16 for 64-bit addressing mode.

The operands *address1* and *address2* may be question marks if:

- the address cannot be resolved because the base or index registers used to compute them have been altered by the instruction
- the operand is ignored by the instruction.

***address3/address4/address5***

Addresses of storage alterations.<sup>2</sup>

***content1***

The contents of the R1 field (RRE format only).<sup>1</sup>

***content2***

The contents of the R2 field (RRE format only).<sup>1</sup>

***vvvv***

A virtual device number.

***vsch***

A virtual subchannel number.

***mnem***

The mnemonic for an instruction. This is a 7-character field in which the mnemonic is left-justified.

***code***

An interruption code number in hexadecimal.

***name***

An interruption name (for example, OPERATION, ADDRESSING) which may be combined with TRANSACTION ABORT, PER or PROGRAM EVENT RECORDER.

***CC n***

The condition code number (0, 1, 2, or 3).

**Note:** The condition code will not always be displayed.

***Gnn***

A general register with altered contents.

***ccccccc***

***ccccccccccccccc***

***ccccccc\_cccccc***

The address generated and loaded by a LOAD ADDRESS or LOAD ADDRESS EXTENDED instruction.<sup>2</sup>

***ss***

A status byte.

***FSA=nnnnnnnn***

***FSA=nnnnnnnnnnnnnnnn***

***FSA=nnnnnnnn\_nnnnnnn***

The failing storage address if valid. If it is not valid, nothing is shown.<sup>2</sup>

***zz***

An altered byte of an EXECUTE instruction, formed by inclusive-ORing the low byte of R1 with the second byte of the executed instruction.

***\*\*\****

A prefix indicating virtual machine interruption.

***\*TN***

A prefix indicating a virtual machine program interruption occurred during transactional execution mode. The address that follows is the start of the aborted transaction when the instruction at the point of the abort is also displayed in the preceding line. When the abort point cannot be determined, the interrupting instruction is not shown, and the address that follows \*TN is the instruction address from the program old PSW. This is either the address after the TBEGIN or the address of the TBEGINC.

## TRACE: Command Responses

->

As a prefix indicating transfer of control, this symbol is present at the beginning of a line whenever the displayed instruction does not immediately follow the last instruction displayed. As a suffix for a successful branch instruction, it precedes the address being transferred to.

-

A suffix used on storage operand addresses to indicate that the address being displayed is one byte past the end of the storage altered by the instruction.

>>

An indication of storage alteration, followed by the operand address.

=

A prefix used with LOAD ADDRESS or LOAD ADDRESS EXTENDED to show that the actual data is shown rather than computing the operand address by base and displacement.

### **AR***reg*

An access register with altered contents.

### **SPACE***owner:space\_name*

identifies the owner's user ID and the name of the address space in which the trace event occurred. It is displayed only for XC virtual machines.

### **ASIT***asit*

identifies the address space identification token of the address space in which the trace event occurred. It is displayed only for XC virtual machines.

Basic Instruction Display (when full trace support is provided)

```
aaaaaaaa mnem xxxxxxxx address1 address2 CC n
aaaaaaaa_aaaaaaaa mnem xxxxxxxx address1
                                address2 CC n
```

Basic Instruction Display (when full trace support is not provided)

```
aaaaaaaa mnem xxxxxxxx CC n
```

Executed Instructions

**Note:** If the target of an EXECUTE instruction modifies a register that was also used to address the target instruction, that is, base or index registers, trace output starting with *address1* and continuing to the end of the trace entry will be incorrect.

```
aaaaaaaa EX xxxxxxxx address1 mnem xxxxxxxx address1 address2 CC n
aaaaaaaa_aaaaaaaa EX xxxxxxxx
                                address1 mnem xxxxxxxx address1 address2 CC n
```

Load Address and Load Address Extended (LA or LAE) Instructions

```
aaaaaaaa mnem xxxxxxxx = ccccccc CC n
aaaaaaaa_aaaaaaaa mnem xxxxxxxx = ccccccc CC n
```

RRE Format Instructions using General Registers

```
aaaaaaaa mnem xxxxxxxx content1 content2 CC n
aaaaaaaa_aaaaaaaa mnem xxxxxxxx content1
                                content2 CC n
```

RRE Format Instructions using Floating Point Registers

```
aaaaaaaa mnem xxxxxxxx CC n
```

Perform-Locked-Operation Instructions

```

aaaaaaaa PLO xxxxxxxx >> address1 address2
                        [address3 address4 address5] CC n
aaaaaaaa_aaaaaaaa PLO xxxxxxxx >> address1
                        address2 [address3 address4]
                        [address5] CC n

```

### RI Format Instructions

```

aaaaaaaa mnem xxxxxxxx CC n

```

### Instructions Altering Storage

```

aaaaaaaa mnem xxxxxxxx >> address1 address2 CC n

```

### Successful Branch Instructions (including SVC and LPSW)

```

aaaaaaaa mnem xxxxxxxx -> tttttttt CC n

```

### Instructions Satisfying GPR Alteration Tracing

```

aaaaaaaa mnem xxxxxxxx address1 address2
CC n Gnn=xxxxxxxx
     Gnn=xxxxxxxx     Gnn=xxxxxxxx     Gnn=xxxxxxxx     ...

```

```

aaaaaaaa_aaaaaaaa mnem xxxxxxxx address1 address2
CC n Gnn=xxxxxxxx_xxxxxxx
     Gnn=xxxxxxxx_xxxxxxx     Gnn=xxxxxxxx_xxxxxxx     Gnn=xxxxxxxx_xxxxxxx

```

### Instructions Satisfying Access Register Alteration Tracing

```

aaaaaaaa mnem xxxxxxxx address1 address2
CC n ARareg=xxxxxxxx
     ARareg=xxxxxxxx     ARareg=xxxxxxxx     ARareg=xxxxxxxx     ...

```

```

aaaaaaaa_aaaaaaaa mnem xxxxxxxx address1 address2
CC n ARareg=xxxxxxxx
     ARareg=xxxxxxxx     ARareg=xxxxxxxx     ARareg=xxxxxxxx     ...

```

### External Interruptions

```

*** aaaaaaaaa EXT code -> tttttttt

```

### Program Interruptions

(Instruction displayed where possible)

```

*** aaaaaaaaa PROG code -> tttttttt name

```

### Transactions Aborted by Program Interruptions

(Instruction displayed where possible)

```

*TN aaaaaaaaaaaaaaaaaa PROG code -> tttttttttttttt

```

### Adapter Interruption Facility Interruptions

```

*** aaaaaaaaa ADAPTER INT -> tttttttt ISC x SCH TYPE y

```

where:

aaaaaaaa = I/O Old PSW (where your program was interrupted)

ttttttt = I/O New PSW (the I/O interruption handler)

ISC x = the Interruption Subclass (bits 2–4 of the  
Host Interruption Identification Word)

## TRACE: Command Responses

SCH TYPE *y* = the Subchannel Type (bits 17–19 of the Host Interruption Identification Word)

Virtual Machine Check Interruptions

(Instruction displayed where possible)

```
*** aaaaaaaaa
MCH -> tttttttt type type ...
      (FSA=nnnnnnnn)
```

### **aaaaaaaa**

identifies the address contained in the second word of the old PSW, if valid, of the machine check. If the PSW instruction address is marked not valid, question marks appear in place of the address.

### **tttttttt**

identifies the address contained in the second word of the new PSW for the machine check.

### **type**

identifies one or more of the machine check codes found in [z/Architecture Principles of Operation](#). The major virtual machine check codes are:

### **FSA=nnnnnnnn**

identifies the FAILING STORAGE ADDRESS if valid. If it is not valid, nothing is shown.

Table 47. Machine Check Codes

Number	Code	Machine Check
0	SD	System damage
1	PD	Instruction processing damage
2	SR	System recovery
3	TD	Interval timer damage
4	CD	Timing facility damage
5	ED	External damage
6	Unused	
7	DG	Degradation
8	W	Warning
9	CP	Channel report pending
10	SP	Service processor damage
11	CK	Channel subsystem damage
12	Unused	
13	Unused	
14	B	Backed up

I/O interruptions

```
*** aaaaaaaaa
I/O vvvv -> tttttttt
SCH vsch PARM xxxxxxxx
```

I/O instructions

```
aaaaaaaa SSCH xxxxxxxx address2
CC 0 SCH vsch DEV vvvv
```

```
CPA xxxxxxxx PARM xxxxxxxx
KEY x FPI xx LPM xx
```

SSCH instructions for Transport Mode I/O may also include the following data structures, if valid:

```
TCW +00000 xxxxxxxx xxxxxxxx ...
TCB +00000 xxxxxxxx xxxxxxxx ...
TSB +00000 xxxxxxxx xxxxxxxx ...
TIDAW xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
DATA +00000 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

**CPA**

identifies the channel program address specifying the absolute address of the first channel command word to be processed on the device.

**PARM**

identifies the interruption parameter to be returned with any interruption for the specified channel command words.

**KEY**

identifies the storage protection key to be used.

**FPI**

identifies the format, prefetch, initial status control byte.

**LPM**

identifies the logical path mask.

**TCW**

the Transport Command Word (Transport Mode only).

**TCCB**

the Transport Command Control Block (Transport Mode only).

**TSB**

the Transport Status Block (Transport Mode only).

**TIDAW**

the Transport Indirect Data Address Word (Transport Mode only).

**DATA**

the data associated with a TIDAW (Transport Mode only).

```
aaaaaaaa TSCH xxxxxxxx address2 CC 0 SCH vsch DEV vvvv
CCWA xxxxxxxx DEV STS xx SCH STS xx CNT xxxx
KEY x FPI xx CC x CTLS xxxx
SENSE DATA: xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
              xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

```
aaaaaaaa TSCH xxxxxxxx address2 CC 1 SCH vsch DEV vvvv
KEY x FPI xx CC x CTLS xxxx
```

**CCWA**

identifies the ending channel command word address when I/O is completed.

**DEV STS**

identifies the unit status for the device.

**SCH STS**

identifies the channel status for the subchannel.

**CNT**

identifies the residual CCW count upon I/O completion.

**KEY**

identifies the key used for storage protection.

**FPI**

identifies the format, prefetch, initial status control byte used. It also indicates a no path available condition and a confirmed I/O condition code 0.

## TRACE: Command Responses

### CC x

identifies the condition code for the requested I/O operation.

### CTLS

identifies the function and activity control bytes.

### SENSE DATA

identifies optional concurrent sense data when concurrent sense is enabled and a unit check has occurred.

```
aaaaaaaa TPI xxxxxxxx address2 CC 1  
SCH xxx DEV xxx PARM xxxxxxxx
```

### SCH

identifies the subchannel requesting the interruption.

### DEV

identifies the subchannel device.

### PARM

identifies the interruption parameter supplied by the virtual machine; it is zero if the interruption is unsolicited.

### Instructions/Condition Codes

```
aaaaaaaa mnem xxxxxxxx address2  
CC n SCH vsch DEV vvvv
```

### Executed I/O Instructions

```
aaaaaaaa EX xxxxxxxx aaaaaaaaa mnem xxxxxxxx address2  
CC n  
SCH vsch DEV vvvv
```

(followed by fields as before.)

### CCW Tracing Display Formats

#### Define Extent CCWs for Mini-DASDs

```
CCW aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh .....  
EXTENT hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
```

#### Define Extent CCWs for Attached DASDs

```
CCW aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh hhhhhhhh  
EXTENT hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
```

#### Locate CCWs for Mini-DASDs

```
CCW aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh .....  
LOCATE gggggggg gggggggg hhhhhhhh hhhhhhhh
```

#### Locate CCWs for Attached DASDs

```
CCW aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh hhhhhhhh  
LOCATE gggggggg gggggggg hhhhhhhh hhhhhhhh
```

#### Locate Record CCWs for Mini-DASDs

```
CCW aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh .....  
LOCATE RECORD hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
```

#### Locate Record CCWs for Attached DASDs





## TRACE: Command Responses

```
CCW      gggggggg gggggggg TIC  aaaaaaaaa
```

### Other CCWs

```
CCW      aaaaaaaaa gggggggg gggggggg  
oooo    hhhhhhhh hhhhhhhh
```

### CCWs with Data Crossing Page Boundaries

```
CCW      aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh .....  
IDAL                                hhhhhhhh  
IDAL                                hhhhhhhh
```

... as many IDALs as needed ...

### CCWs with User IDALs

```
CCW      aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh .....  
IDAL                                gggggggg  
IDAL                                gggggggg hhhhhhhh  
...                                     ...
```

### CCWs with User Format-2 IDALs

```
CCW      aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh .....  
IDAL                                gggggggg_gggggggg hhhhhhhh_hhhhhhhh  
IDAL                                gggggggg_gggggggg hhhhhhhh_hhhhhhhh  
...                                     ...
```

### CCWs with Virtual Channel IDAWs

```
CCW  
IDAW    gggggggg DATA gggg
```

### CCWs with Virtual Channel Format-2 IDAWs

```
CCW  
IDAW    gggggggg_gggggggg DATA gggggggg
```

### Data Transfer over Eight Bytes with Virtual Channel Simulator

```
CCW  
hhhhhhh hhhhhhh  
hhhhhhh hhhh
```

### Suspend with Virtual Channel Simulator

```
CCW      gggggggg gggggggg ***SUSPENSION***
```

### CCWs with Virtual Channel MIDAWs (Fully Simulated CCWs)

```
CCW  
MIDAW   gggggggg_gggggggg DATA gggggggg
```

### CCWs with User MIDAL (CCWs Sent to Hardware)

```
CCW      aaaaaaaaa gggggggg gggggggg oooo hhhhhhhh .....  
MIDAW   gggggggg_gggggggg gggggggg_gggggggg hhhhhhhh_hhhhhhhh  
MIDAW   gggggggg_gggggggg gggggggg_gggggggg hhhhhhhh_hhhhhhhh  
...     ...
```

## TRACE AIF

---



### Authorization

Privilege Class: G

### Purpose

Use TRACE AIF to trace the interruptions generated by the Adapter Interruption Facility.

### Operands

#### *options*

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

The FROM option is not valid for TRACE AIF.

### Responses

For more information, see [“Usage Notes” on page 1946](#).

## TRACE ALL

---



### Authorization

Privilege Class: G

### Purpose

Use TRACE ALL to trace the following:

- All program, external, and I/O interrupts
- All instructions executed.

### Operands

#### *options*

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

### Usage Notes

1. Because ALL encompasses system events in your entire address range, the PSWA option has no function if used with TRACE ALL. If an address range is specified by the PSWA operand, it is ignored.
2. Each TRACE ALL command replaces an active TRACE ALL command.
3. For more information, see TRACE [“Usage Notes” on page 1946](#).

### Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE APPEND

---

```
▶ TRace — APPEND — name ▶
```

### Authorization

Privilege Class: G

### Purpose

Use TRACE APPEND to merge the trace traps from two trace sets into one.

### Operands

#### *name*

specifies the trace set whose traps are to be merged into the current trace set.

### Usage Notes

1. The TRACE APPEND command causes all trace traps from the named trace set to be added to the current trace set.

If a trap exists in both trace sets with the same identifier and the trap was named using the ID option on the TRACE command, the trap in the named set replaces the trap in the current trace set.

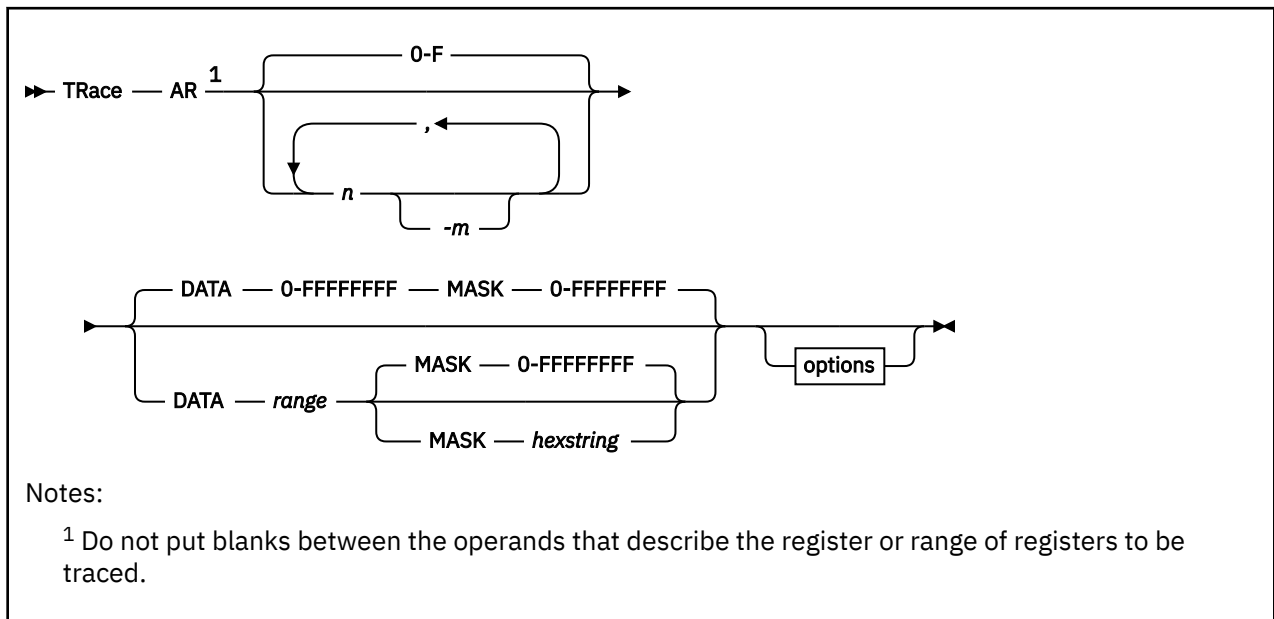
If two traps have the same identifier but were not named with the ID option (for example, the identifier was assigned by the system), both traps remain, and the trap from the named trace set is assigned a new identifier.

2. The current trace set cannot be appended to itself.
3. For more information, see TRACE [“Usage Notes”](#) on page 1946.

### Responses

For response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE AR



## Authorization

Privilege Class: G

## Purpose

Use TRACE AR to trace changes to the virtual machine access registers. You can trace all changes, or only those that cause a particular register or set of access registers to assume a specified value.

## Operands

### AR...

is the access register or range of access registers whose alterations you want to trace. If you are specifying a range of access registers, use a hyphen between the first register number and last register number in the range.

### DATA range

specifies a data value or range of data values to be checked to see if any of the changed registers assume a value within the range. The range may be a single value or a pair of values separated by a period (.) or a dash (-). A dash denotes a starting and ending range. A period denotes a starting range for a byte count. Range values are viewed as unsigned 32-bit integers. If no data range is specified for general-register alteration tracing, 0-FFFFFFFF is assumed. Wrap-around data ranges are not allowed.

### MASK string

specifies a hexadecimal string that forms a mask for data comparison. The MASK string is logically applied to both the register and the data range before a comparison is made.

### options

For an explanation of the options, see [“TRACE: Options”](#) on page 1949.

## Usage Notes

1. You may specify the register numbers in either decimal or hexadecimal. Wrap-around ranges (that is, ranges whose ending register is less than the starting register) are permitted. For example: `trace`

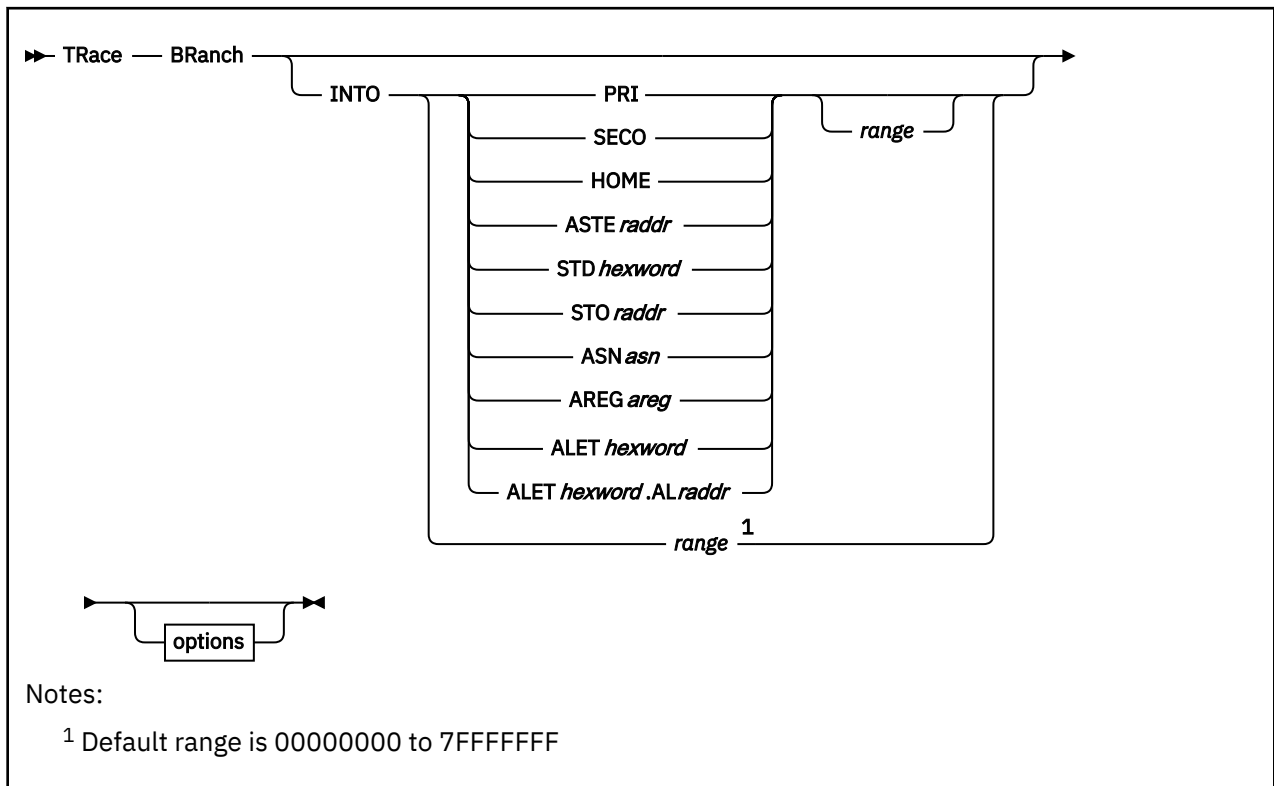
AR10-2 is a valid range; it establishes tracing of alterations to registers 10, 11, 12, 13, 14, 15, 0, 1, and 2.

2. Your virtual machine runs very slowly when you enter the TRACE AR command.
3. For more information, see TRACE [“Usage Notes” on page 1946](#).

## Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE BRANCH



### Authorization

Privilege Class: G

### Purpose

Use TRACE BRANCH to trace successful branch instructions, including load PSW and SVC instructions that do not load a wait PSW.

### Operands

#### PRI

restricts tracing to events in the guest's current primary address space as pointed to by guest control register 1.

#### SECO

restricts tracing to events in the guest's current secondary address space as pointed to by guest control register 7.

#### HOME

restricts tracing to events in the guest's current home address space as pointed to by guest control register 13.

#### ASTE*raddr*

specifies that an ASN second table entry (ASTE) is to be used to locate the address space to be traced. The operand *raddr* is the guest real address of the ASTE that contains the segment table origin (STO) of the address space to be traced.

#### STD*hexword*

specifies that a segment table designation (STD) is to be used to locate the address space to be traced. The STD contains the segment table origin (STO) of the address space to be traced.



**STOraddr**

specifies that a segment table origin (STO) is to be used to locate the address space to be traced. The operand *raddr* is the segment table origin of the address space to be traced and is a guest real address.

**ASNasn**

specifies that an address space number (ASN) is to be used to locate the address space to be traced. The specified ASN may be one to four hexadecimal digits long. The ASN is used in conjunction with guest ASN translation tables to locate the STO of the address space to be traced.

**AREGareg**

specifies that access register translation is to be used to locate the address space to be traced. The access register (*areg*) is specified as one hexadecimal or two decimal digits. The ALET contained within the access register is used in conjunction with the guest access register-translation tables to locate the STO of the address space to be traced.

**ALETheyword**

specifies that access register translation is to be used to locate the address space to be traced. The ALET is used in conjunction with the guest access register-translation tables to locate the STO of the address space to be traced.

**ALETheyword.ALraddr**

restricts tracing to events in an address space using the specified access list entry token (ALET) and access list (AL). The ALET and AL are used in conjunction with the guest access register-translation tables to locate the STO of the address space to be traced.

**INTO range**

permits a trap to be taken if the destination of a successful branch instruction is within the specified range.

You may specify the range as a single address or a pair of addresses separated by a period (.) or a dash (-). A dash denotes a starting and ending range; a period denotes a starting range for a byte count. A branch destination is considered to be within the range if any byte is within the range.

The addresses are real or virtual as determined by the setting of the dynamic address translation (DAT) bit in the PSW at the time the event is detected. For BRANCH tracing, the address or range of addresses can be odd or outside of the address space; the event is detected after the branch instruction, but before any attempt is made to fetch the next instruction.

**options**

For an explanation of options, see [“TRACE: Options” on page 1949](#).

**Usage Notes**

1. Your virtual machine runs very slowly after you enter the TRACE BRANCH instruction.
2. While TRACE BRANCH is active, a record of branch instructions is maintained in the traceback table. You can use the TRACE TABLE command to display the traceback table.
3. Guest control register values and translation table contents define the address space structure of the virtual machine. Program execution (and the STORE command) can change that structure over time. The TRACE command operands that specify a guest address space to be traced by a trace trap use the current guest control register and translation table values to determine the segment table origin (STO) of the address space to be traced (at the time the TRACE command is issued). The STO is contained in the trace trap and can be examined using the QUERY TRACE command—that is, you specify an address space based on the state of your virtual machine at the time of the TRACE command that establishes the trace trap.

A trace trap generated with a specific address space specification (STO) will always identify that address space using the STO generated from the control registers, the translation tables, or both. If your program execution changes the control register values and translation table contents to represent a different address space structure, there is *no* change to the trace traps. The trace traps remain in effect with the address space specified at the time of the TRACE command execution.

## TRACE BRANCH

4. The operands HOME, AREG*areg*, ALE*hexword*, and ALE*hexword*.AL*raddr* are valid only if the virtual machine is an ESA or XA virtual machine.
5. When the operand ALE*hexword*.AL*raddr* is used, the primary list bit in the ALET is ignored. Because this bit will choose between access lists, and the access list has been explicitly specified, its function is ignored.
6. The operands PRI, SECO, HOME, ASTE*raddr*, STD*hexword*, ASN*asn*, AREG*areg*, ALE*hexword*, and ALE*hexword*.AL*raddr* use the access-register-translation (ART), the ASN translation and the dynamic-address-translation (DAT) processes. These contain special case translation processes including the following:
  - ALET 0 always indicates the current primary address space designated by the STD in guest control register 1, regardless of the current contents of the dispatchable unit access list.
  - ALET 1 always indicates the current secondary address space designated by the STD in guest control register 7, regardless of the current contents of the dispatchable unit access list.
  - Access register 0 is always treated as containing ALET 0. The contents of access register 0 are not used during ART processing.
7. A traced address space is identified by its segment table origin (STO). The STD operand is provided for consistency with the DISPLAY, DUMP, and STORE commands. The STO is extracted from the STD provided and all other STD fields are ignored.
8. If an ESA mode guest specifies an address space option from which an ASN can be derived, then trace will trap events in all subspaces which are members of the group identified by the ASN whenever possible.

An ASN can be derived from the following address space options.

- PRI
  - SECO
  - ASN
  - AREG0
  - AREG*x*, where the ALET in AR*x* is zero or one
  - ALET*n*, where *n* is zero or one
9. For more information, see TERMINAL [“Usage Notes” on page 1946](#).

## Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE CLEAR

---



### Authorization

Privilege Class: G

### Purpose

Use TRACE CLEAR to delete all traps in a named trace set.

**Note:** The trace set itself is deleted if it is not active, and if it is not found in the RETURN list constructed by TRACE CALL commands.

### Operands

#### *name*

deletes all traps in the named trace set. If you do not specify a name, the active trace set is cleared.

### Usage Notes

1. For usage information, see TRACE [“Usage Notes”](#) on page 1946.

### Responses

For response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE COUNT

---

►► TRace — COunt ◄◄

### Authorization

Privilege Class: G

### Purpose

Use TRACE COUNT to count the successful trace events in your virtual machine.

### Usage Notes

1. The TRACE COUNT function lets you count the number of successful trace events in the current trace set. While TRACE COUNT is active, no other trace output is produced. The virtual machine does not stop after each event, even if you specify NORUN.
2. You can use the TRACE END COUNT command to end TRACE COUNT. However, when you enter any valid TRACE command, the system displays the value of the current count, and TRACE COUNT terminates.

If you enter TRACE COUNT when COUNT is already active, the system displays the current count, and the count resumes from 0.

When the TRACE COUNT reaches 16 million, the system displays that count and resumes from 0.

3. The current COUNT is also included in the QUERY TRACE response.
4. For more information, see TRACE [“Usage Notes”](#) on page 1946.

### Responses

Response 1:

```
Trace count is nnnnnnnn
```

is returned before the display of trace traps if TRACE COUNT is active.

**nnnnnnnn**

represents the successful trace events in the current trace set.

**Note:** For other response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE DELETE

---

```
▶▶ TRace — DElete — ident ▶▶
```

### Authorization

Privilege Class: G

### Purpose

Use TRACE DELETE to delete a named trace trap.

### Operands

#### DElete

deletes an event from the current set.

#### *ident*

identifies a trace trap describing the event; it must be one to four characters long. You cannot use the word ALL as an identifier; on the TRACE DELETE command, this option is not valid.

### Usage Notes

1. For more information, see TRACE [“Usage Notes”](#) on page 1946.

### Responses

For response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE END



### Authorization

Privilege Class: G

### Purpose

Use TRACE END to delete all trace traps and all trace sets.

**Note:** No return list remains.

### Operands

#### ALL

deletes all trace traps and trace sets. ALL is the default.

#### COUNT

requests the display of the current value of the TRACE COUNT and the termination of TRACE COUNT.

### Usage Notes

1. For usage information, see TRACE [“Usage Notes”](#) on page 1946.

### Responses

Response 1:

```
Trace ended
```

is returned if tracing is in effect when you enter TRACE END ALL. If tracing is not in effect, the system displays no message.

Response 2:

```
Trace count is nnnnnnnn
Trace count ended
```

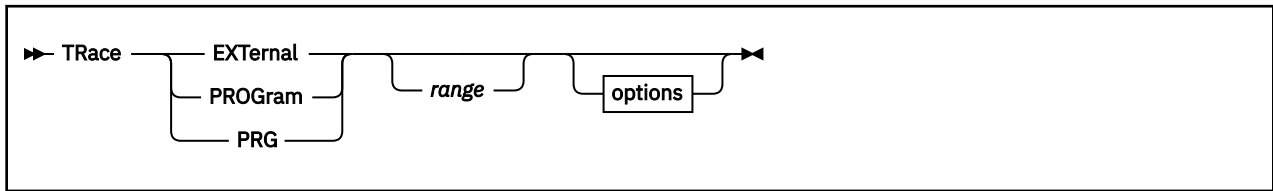
is returned if TRACE COUNT is active when you enter TRACE END COUNT.

#### **nnnnnnnn**

identifies the number of successful trace events in the current trace set.

**Note:** For other response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE EXTERNAL/PROGRAM



### Authorization

Privilege Class: G

### Purpose

Use TRACE EXTERNAL or TRACE PROGRAM to trace external or program interrupts, or a range of external or program interrupts.

**Note:** Where possible, the instruction that caused the program interrupt is displayed also.

### Operands

#### *range*

is a hexadecimal range of external or program interrupts. You can express this range as a single value, a pair of values separated by a hyphen (-) or an address followed by a period (.) and followed by a byte count. Wrap-around ranges are not allowed.

#### *options*

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

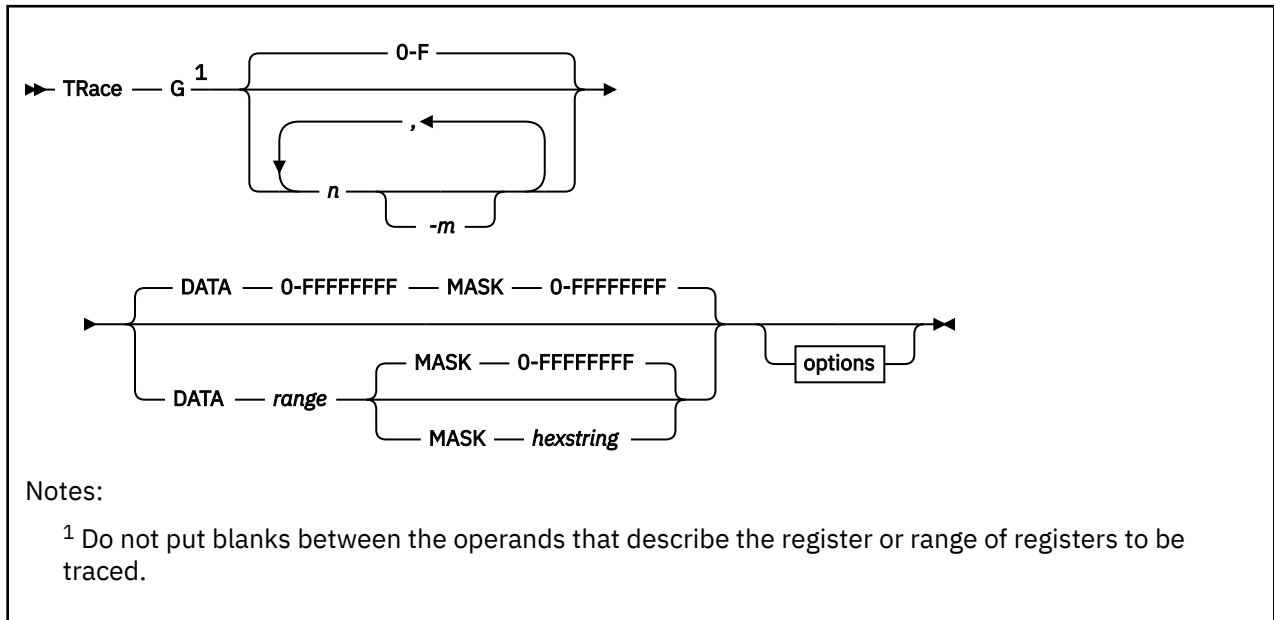
### Usage Notes

1. Because a PER event may occur concurrently with another program interruption, a TRACE PROG trap may hit on either a PER event alone, a PER event with another program interruption, or a program interruption alone. For example: TRACE PROG 80 will trap all PER events, whether or not they have another program interruption occurring concurrently. TRACE PROG 4 will trap all PROG 4 interruptions whether or not they have a PER event occurring concurrently. TRACE PROG 84 will trap a PROG 84 (protection exception with a PER event) only.
2. For additional usage information, see TRACE [“Usage Notes” on page 1946](#).

### Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE G

**Authorization**

Privilege Class: G

**Purpose**

Use TRACE G to trace changes in your virtual machine general purpose registers.

**Note:** You can trace all changes or only those that cause a particular register, or set of registers, to assume a specified value.

**Operands****G...**

identifies the general register or range of general registers whose alterations you want to trace. If you are specifying a range of general purpose registers, use a hyphen between the first register number and last register number in the range.

You may specify the register numbers in either decimal or hexadecimal. Wrap-around ranges (that is, ranges whose ending register is less than the starting register) are permitted. For example, **trace G10-2** is a valid range; it establishes tracing of alterations to registers 10, 11, 12, 13, 14, 15, 0, 1, and 2.

**DATA range**

specifies a data value or range of data values to be checked to see if any of the altered registers assume a value within the range. The range may be a single value or a pair of values separated by a period (.) or a hyphen (-). A hyphen denotes a starting and ending range. A period denotes a starting range for a byte count. Range values are viewed as unsigned 32-bit integers. If no data range is specified for general register alteration tracing, 0-FFFFFFFF is assumed. Wrap-around data ranges are not allowed.

**MASK hexstring**

specifies a hexadecimal string that forms a mask for data comparison. The MASK string is logically applied to both the register and the data range before a comparison is made.

**options**

For an explanation of the options, see [“TRACE: Options” on page 1949](#).



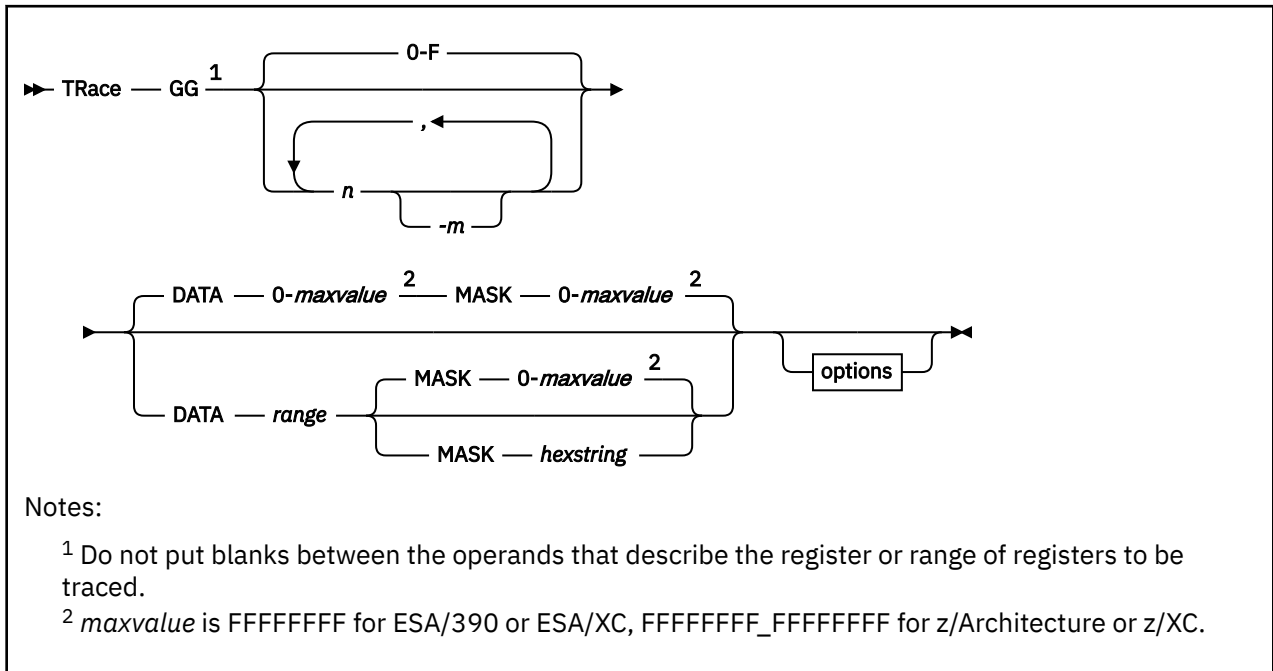
## Usage Notes

1. Your virtual machine runs very slowly after you enter the TRACE G command.
2. For more information, see TRACE [“Usage Notes”](#) on page 1946.

## Responses

For response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE GG



### Authorization

Privilege Class: G

### Purpose

Use TRACE GG to trace changes in your virtual machine general purpose registers. TRACE GG is valid in all modes. If the virtual machine is in ESA/390 or ESA/XC mode it will operate on a four-byte register. If it is in z/Architecture or z/XC mode, it will operate on an eight-byte register.

If the data with the mask applied results in a value greater than FFFFFFFF, a TRACE GG trap will never hit unless the guest is in z/Architecture or z/XC mode.

If the DATA and MASK options are not specified, TRACE GG will trap all alterations of a register regardless of the virtual machine mode.

**Note:** You can trace all changes or only those that cause a particular register, or set of registers, to assume a specified value.

### Operands

#### GG...

identifies the general register or range of general registers whose alterations you want to trace. If you are specifying a range of general purpose registers, use a hyphen between the first register number and last register number in the range.

You may specify the register numbers in either decimal or hexadecimal. Wrap-around ranges (that is, ranges whose ending register is less than the starting register) are permitted. For example, `trace GG10-2` is a valid range; it establishes tracing of alterations to registers 10, 11, 12, 13, 14, 15, 0, 1, and 2.

#### DATA range

specifies a data value or range of data values to be checked to see if any of the altered registers assume a value within the range. The range may be a single value or a pair of values separated by a

period (.) or a hyphen (-). A hyphen denotes a starting and ending range. A period denotes a starting range for a byte count.

In ESA/390 or ESA/XC mode, range values are viewed as eight hexadecimal digits. Wrap-around data ranges are not allowed. If no data range is specified for general register alteration tracing, 0-FFFFFFFF is assumed.

In z/Architecture or z/XC mode, range values are viewed as 16 hexadecimal digits. If no data range is specified for general register alteration tracing, 00000000-FFFFFFFF\_FFFFFFFF is assumed. Wrap-around data ranges are not allowed. An optional delimiter (underscore) can be used to separate the two words of a 16 hexadecimal digit input value. If the underscore delimiter is used there must be exactly eight hexadecimal digits to the right of the delimiter. Leading zeroes on the word to the left of the delimiter can be omitted.

**MASK hexstring**

specifies a hexadecimal string that forms a mask for data comparison. The MASK string is logically applied to both the register and the data range before a comparison is made.

In ESA/390 or ESA/XC mode, the mask string is eight hexadecimal digits. The default is FFFFFFFF.

In z/Architecture or z/XC mode, the mask string is 16 hexadecimal digits. The default is FFFFFFFF\_FFFFFFFF.

**options**

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

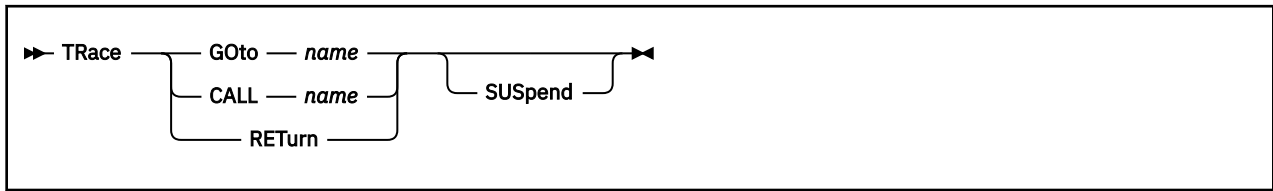
**Usage Notes**

1. Your virtual machine runs very slowly after you enter the TRACE GG command.
2. For more information, see TRACE [“Usage Notes” on page 1946](#).

**Responses**

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE GOTO/CALL/RETURN



### Authorization

Privilege Class: G

### Purpose

Use TRACE GOTO, TRACE CALL, and TRACE RETURN to activate TRACE sets by name.

### Operands

#### GOTO *name*

causes a trace set by the given name to become active. Specify *name* as a unique 1- to 8-character identifier. If no trace set exists by that name, a new, empty set is created. No program degradation occurs from an empty (null) trace set. The previous trace set is unchanged but no longer active. If empty, it may be deleted automatically.

You can use TRACE GOTO as the first command in a tracing session to establish the name of an initial trace set. If traps are created without issuing a TRACE GOTO, the initial trace set containing them is named INITIAL.

#### CALL *name*

causes a trace set by the given name to become active. If no trace set exists by that name, a new, empty set is created. No program degradation occurs from an empty (null) trace set. The previous trace set is unchanged and is placed in a return list on a last-in, first-out basis.

You can enter the QUERY TRACE RETURNS command to display a list of trace sets currently in the return list; the order is from most to least recent. Recursive calling is not allowed; it is not permitted to call any trace set that has itself called a trace set.

TRACE CALL is designed to be used with TRACE RETURN.

#### RETurn

causes the first trace set on the return list to be removed from the list and become active. The previous trace set is unchanged but no longer active; if empty, the previous trace set is deleted automatically.

#### SUSpend

causes the trace set being loaded to have no effect until the next TRACE GOTO, TRACE CALL, or TRACE RETURN occurs. You can enter TRACE commands to place traps for trace events in the set without affecting the program that is currently in operation. A suspended trace set causes no program degradation.

### Usage Notes

1. You can construct trace sets by coding a program that runs in your virtual machine. The program would retrieve commands from storage and pass them to the control program using DIAGNOSE instructions.

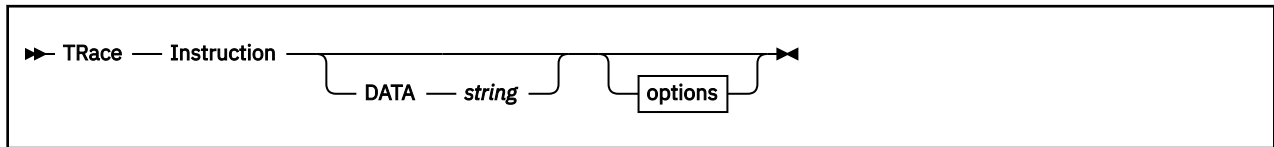
In most cases, the commands are to be applied to a subsequent program that is not yet in operation. You can use the SUSPEND option to ensure that the trace set has no effect until the next TRACE GOTO, TRACE CALL, or TRACE RETURN occurs.

2. For more information, see TRACE [“Usage Notes”](#) on page 1946.

## Responses

For response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE INSTRUCTION



### Authorization

Privilege Class: G

### Purpose

Use TRACE INSTRUCTION to trace instructions in your virtual machine, regardless of the success of command execution.

### Operands

#### **DATA *string***

specifies a string of data to be compared against the instruction data.

#### ***options***

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

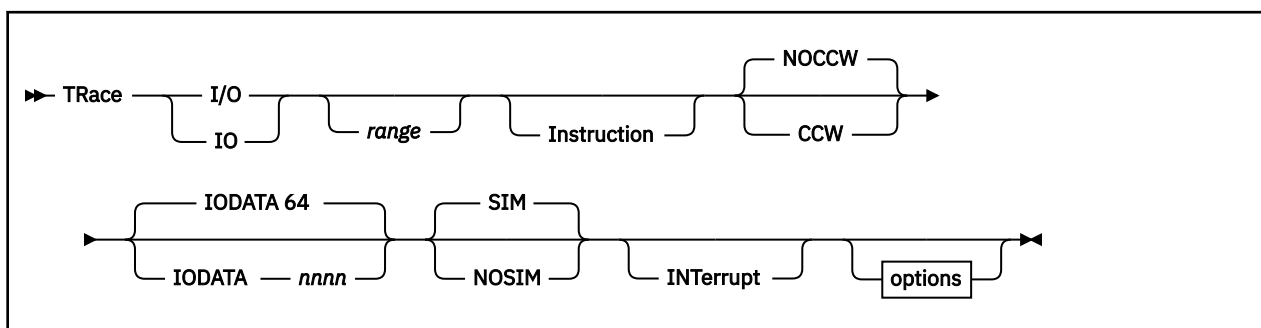
### Usage Notes

1. If you use TRACE INSTRUCTION in conjunction with the PSWA option, only instructions within the range of the address specified by PSWA are traced.
2. You may use the DATA option to limit tracing to certain instructions or sequences of instructions. For example, `trace instr data 183f` causes the tracing of all LR 3,15 instructions; the command `trace instr data 183f50` traces all LR 3,15 instructions only if they are followed by a STORE (ST) instruction.
3. For more information, see TRACE [“Usage Notes” on page 1946](#).

### Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE I/O



### Authorization

Privilege Class: G

### Purpose

Use the **TRACE I/O** command to control virtual machine tracing of I/O events.

### Operands

#### *range*

identifies a range of devices. The range may be expressed as a single device number or pair of device numbers separated by a hyphen (-), or a starting device number followed by a period (.) and followed by a number representing the number of consecutive device addresses to be traced. Wrap-around ranges are not allowed.

#### **Instruction**

specifies tracing of I/O instructions for a TRACE I/O command. If neither INSTRUCTION nor CCW nor INTERRUPT is specified, INSTRUCTION and INTERRUPT are assumed.

#### **NOCCW**

specifies that CCWs are not to be displayed. NOCCW is assumed if CCW is not specified.

#### **CCW**

specifies that CCWs are to be displayed with RIO, SIO, SIOF, SSCH, and RSCH instructions that successfully start I/O.

#### **IODATA nnnn**

For Command Mode CCWs and Transport Mode structures handled by the Virtual Channel Simulator, IODATA specifies the number of data bytes to be displayed for each CCW. *nnnn* can be a value from 0 - 3800, with a default of 64. If *nnnn* is greater than 64, TRACE I/O will display *nnnn* bytes of data following the CCW and prefixed with a hexadecimal offset:

```
cp trace io 500 inst int ccw iodata 64
... (other SSCH / IO INT trace output) ...
VDEV 0500 CCW 72200080 00E58968 STS 00
      DATA +000 D0010600 F0F0F1F7 F3F2F0F0 F5C9C2D4
      DATA +010 E5D4F0F0 F0F0F0F0 F0F0F0F0 F0F00200
      DATA +020 D0020000 F0F0F1F7 F3F1F0F0 F5C9C2D4
      DATA +030 E5D4F0F0 F0F0F0F0 F0F0F0F0 F0F00200 STS 0C
```

Transport Mode I/O data is displayed in the following format:

```
tr io 6c7 ccw int instr iodata 128
SSCH / IO INT trace output) ...
VDEV 6C7
  TCW +00000 00000000 004E0000 00000000 00000000
  TCW +00010 00000000 00300000 00000000 00010A40
  TCW +00020 00000000 00010800 00000000 0000C000
  TCW +00030 00000000 00000000 00000000 00000000
  TCCB +00000 7F000000 00000054 1FFE0000 00000000
```

## TRACE I/O

```
TCCB +00010 EA000040 0000C000 01800000 2D000000
TCCB +00020 00000000 40C01000 0000004A 00010000
TCCB +00030 0002000E 00000000 00000000 2000000C
TCCB +00040 00000000 0CE8A601 00010000 00010000
TCCB +00050 0100C000 00000000 00000000 0000C004
VDEV 6C7
TSB +00000 20D10000 00000000 00000000 000001FC
TSB +00010 00000000 00000000 00000000 00000000
TSB +00020 00000000 00000000 00000000 00000000
TSB +00030 00000000 00000000 00000000 00000000
DATA +00000 00000000 00000000 00000000 00000000 *.....*
DATA +00010 00000000 00000000 00000000 00000000 *.....*
DATA +00020 00000000 00000000 00000000 00000000 *.....*
DATA +00030 00000000 00000000 00000000 00000000 *.....*
DATA +00040 00000000 00000000 00000000 00000000 *.....*
DATA +00050 00000000 00000000 00000000 00000000 *.....*
DATA +00060 00000000 00000000 00000000 00000000 *.....*
DATA +00070 00000000 00000000 00000000 00000000 *.....*
0000BF80 BYTES NOT DISPLAYED
```

### **SIM**

specifies normal simulation of instructions. If neither SIM nor NOSIM is specified, SIM is assumed.

### **NOSIM**

specifies suppression of instruction simulation. Program interrupts for guest PER events are also suppressed. By using NOSIM and the CMD option, you can define a CP command or set of CP commands to be processed in place of the suppressed instruction. The PSW is left pointing to the next sequential instruction.

### **INTerrupt**

specifies tracing of I/O interrupts for a TRACE I/O command. If neither INSTRUCTION nor CCW nor INTERRUPT is specified, INSTRUCTION and INTERRUPT are assumed.

### **options**

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

### **Usage Notes**

1. For z/VM I/O, the device range applies to the device number, and not the subchannel number.
2. I/O interrupt addresses are not subject to the instruction range (if any) specified by PSWA, FROM, or RANGE.
3. For more information, see TRACE [“Usage Notes” on page 1946](#).

### **Responses**

For response information, see [“TRACE: Command Responses” on page 1954](#).



## TRACE MCH

---



### Authorization

Privilege Class: G

### Purpose

Use TRACE MCH to intercept machine check interrupts.

### Note:

1. Where possible, the instruction that caused the program interrupt will be displayed also.
2. If you enter TRACE MCH and a machine check occurs, the tracing facility saves information from an operation preceding the machine check.

### Operands

#### *options*

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

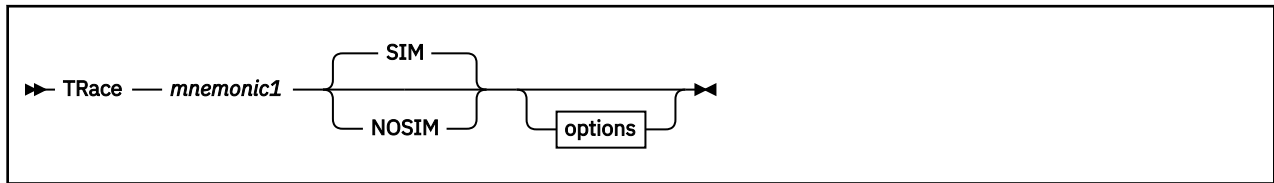
### Usage Notes

1. For usage information, see TRACE [“Usage Notes” on page 1946](#).

### Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE mnemonic1



### Authorization

Privilege Class: G

### Purpose

Use TRACE *mnemonic1* to trace the execution of specific assembler language instructions.

### Operands

#### *mnemonic1*

must be one of the following standard assembler opcode mnemonics:

BSA	PC	SSK
BSG	PFMF	SSKE
CONCS	PGIN	SSM
CSP	PGOUT	STAP
CSPG	PR	STCPS
DISCS	PT	STCRW
EPSW	PTI	STCTG
ESSA	PTLB	STCTL
IDTE	RCHP	STFL
IPTE	RDD	STFLE
ISK	RDP	STIDC
ISKE	RRB	STIDP
IUCV	RRBE	STSI
LASP	SAL	STNSM
LCTL	SCHM	STOSM
LCTLG	SCK	STPX
LPSW	SERV	TB
LPSWE	SIE	TPROT
LPSWEY	SIGP	WRD
PALB	SPX	

Tracing is limited to those instructions that do not produce an operation or privileged operation exception.

The IUCV operand traces both IUCV and APPC/VM macro functions.

**Note:** For information on the [IUCV Overview](#) see *z/VM: CP Programming Services*.

#### **SIM**

specifies normal simulation of instructions. If neither SIM nor NOSIM is specified, SIM is assumed.

#### **NOSIM**

specifies suppression of instruction simulation. Program interrupts for guest PER events are also suppressed. By using NOSIM and the CMD option, you can define a CP command or set of CP commands to be processed in place of the suppressed instruction. The PSW is left pointing to the next sequential instruction.

You can embed instructions in your assembled code as breakpoints and use TRACE *mnemonic1* to stop execution, or you can use TRACE *mnemonic1* to stop program execution before a questionable instruction is executed. The PSW is left pointing to the next sequential instruction.

**options**

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

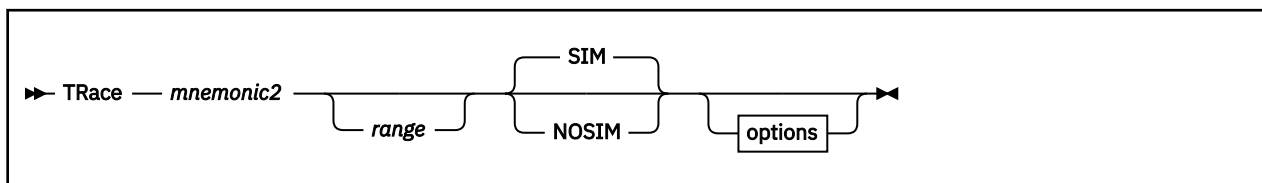
**Usage Notes**

1. When you are tracing mnemonics and an instruction causes an operation exception, the instruction is not displayed.
2. For more information, see TRACE [“Usage Notes” on page 1946](#).

**Responses**

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE mnemonic2



### Authorization

Privilege Class: G

### Purpose

Use TRACE *mnemonic2* to trace specific I/O instructions within a device range.

### Operands

#### *mnemonic2*

must be one of the following standard assembler opcode mnemonics:

CLRCH	MSCH	SSCH
CLRIO	RIO	STSCH
CSCH	RSCH	TCH
EQBS	SIGA	TIO
HDV	SIO	TPI
HIO	SIQF	TSCH
HSCH	SQBS	XSCH

Tracing is limited to those instructions that do not produce an operation or privileged operation exception.

#### *range*

is a range of devices. The range may be expressed as:

- a single device number (100)
- a pair of device numbers separated by a hyphen (100-103)
- device number followed by a period and the number of consecutive device addresses to be traced (100.4).

Wrap-around ranges are not allowed.

#### **SIM**

specifies normal simulation of instructions. If neither SIM nor NOSIM is specified, SIM is assumed.

#### **NOSIM**

specifies suppression of instruction simulation. Program interrupts for guest PER events are also suppressed. By using NOSIM and the CMD option, you can define a CP command or set of CP commands to be processed in place of the suppressed instruction. The PSW is left pointing to the next sequential instruction.

#### *options*

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

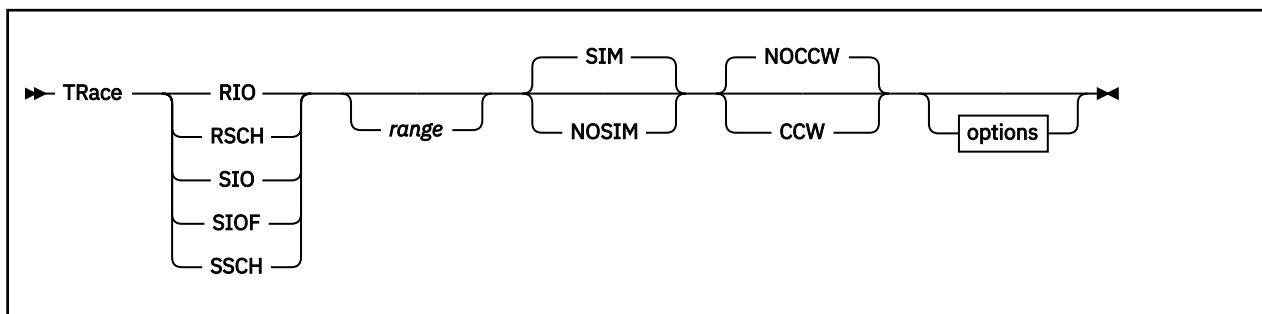
### Usage Notes

1. For usage information, see TRACE [“Usage Notes” on page 1946](#).

### Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE RIO/RSCH/SIO/SIOF/SSCH



### Authorization

Privilege Class: G

### Purpose

Use TRACE RIO, TRACE RSCH, TRACE SIO, TRACE SIOF, or TRACE SSCH to trace RIO, RSCH, SIO, SIOF, or SSCH instructions for all I/O devices, or for a range of devices.

**Note:** Tracing is limited to those instructions that do not produce an operation or privileged-operation exception.

### Operands

#### *range*

is a hexadecimal range of device numbers.

A range may be expressed as:

- A single device number
- A device number followed by a period, followed by a number that represents the number of consecutive device addresses to be traced
- A pair of device numbers separated by a period (.), or a hyphen (-).

Wrap-around ranges are not allowed.

#### **SIM**

specifies normal simulation of instructions. If neither SIM nor NOSIM is specified, SIM is assumed.

#### **NOSIM**

specifies suppression of instruction simulation. Program interrupts for guest PER events are also suppressed. By using NOSIM and the CMD option, you can define a CP command or set of CP commands to be processed in place of the suppressed instruction. The PSW is left pointing to the next sequential instruction.

#### **NOCCW**

specifies that CCWs are not displayed. If you specify neither CCW or NOCCW, NOCCW is assumed.

#### **CCW**

specifies that CCWs be displayed with the SIO, SIOF, SSCH, RIO, and RSCH instructions that successfully start I/O.

#### *options*

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

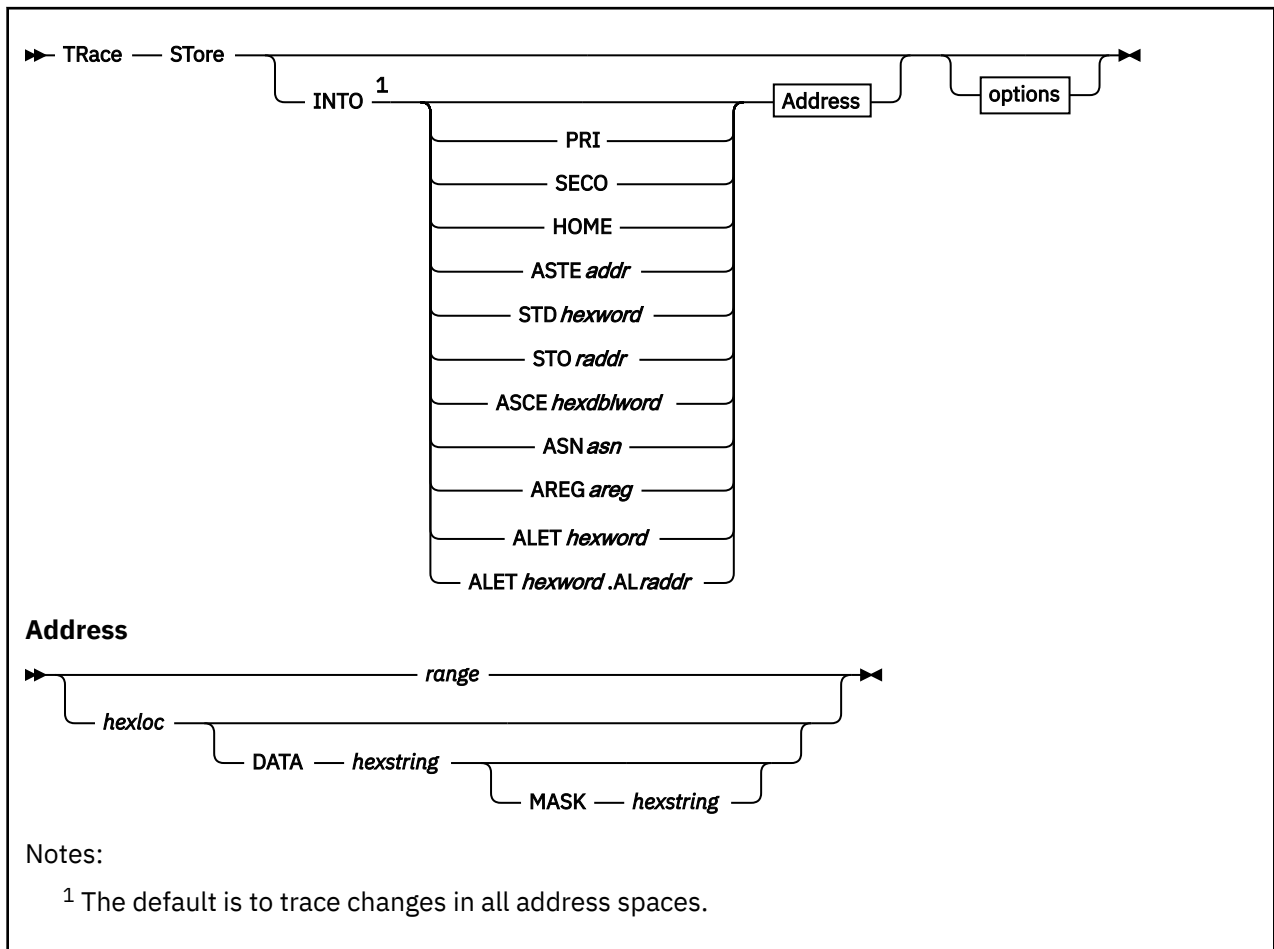
### **Usage Notes**

1. RIO, RSCH, SIO, SIOF, and SSCH instructions that cause an operation or privileged operation exception are not displayed.
2. For usage information, see TRACE [“Usage Notes” on page 1946](#).

### **Responses**

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE STORE (ESA/390, z/Architecture)



### Authorization

Privilege Class: G

### Purpose

Use TRACE STORE (ESA/390, z/Architecture) to trace processor alterations made to virtual storage for an ESA/390 or z/Architecture virtual machine. You can trace all alterations, or only those alterations causing a specified set of logical locations to assume a specified value.

There is another version of the TRACE STORE command for ESA/XC and z/XC virtual machines. See [“TRACE STORE \(ESA/XC, z/XC\)”](#) on page 1995.

### Operands

#### INTO

restricts tracing to a specific storage location or range of locations. Unless an address space is specified, changes to storage are traced in all address spaces.

#### PRI

restricts tracing to events in the guest's current primary address space as pointed to by guest control register 1.

#### SECO

restricts tracing to events in the guest's current secondary address space as pointed to by guest control register 7.

**HOME**

restricts tracing to events in the guest's current home address space as pointed to by guest control register 13.

**ASTERaddr**

specifies that an ASN second table entry (ASTE) be used to locate the address space to be traced. The operand *raddr* is the guest real address of the ASTE that contains the segment table origin (STO) of the address space to be traced.

**STDhexword**

specifies that a segment table designation (STD) be used to locate the address space to be traced. The STD contains the segment table origin (STO) of the address space to be traced.

**STOraddr**

specifies that a segment table origin (STO) be used to locate the address space to be traced. The operand *raddr* is the segment table origin of the address space to be traced and is a guest real address.

**ASCEhexdblword**

specifies that the address space control element (ASCE) be used to locate the address space to be traced. The guest must be in z/Architecture mode for INTO ASCE to be valid. The ASCE option must be followed by 16 hexadecimal digits (no underscore allowed). To specify a range after that you need a blank between the ASCE value and the range:

```
TRACE STOR INTO ASCE1234567812345678 0.0007FFFFFF
```

**ASNasn**

specifies that an address space number (ASN) be used to locate the address space to be traced. The specified ASN may be 1 to 4 hexadecimal digits. The ASN is used in conjunction with guest ASN translation tables to locate the STO of the address space to be traced.

**AREGareg**

specifies that access register translation be used to locate the address space to be traced. The access register (*areg*) is specified as 1 hexadecimal or 2 decimal digits. The access list entry token (ALET) within the access register is used in conjunction with the guest access register translation tables to locate the STO of the address space to be traced.

**ALETheyword**

specifies that access register translation be used to locate the address space to be traced. The access list entry token (ALET) is used in conjunction with the guest access register translation tables to locate the STO of the address space to be traced.

**ALETheyword.ALraddr**

restricts tracing to events in an address space using the specified access list entry token (ALET) and access list (AL). The ALET and AL are used in conjunction with the guest access register translation tables to locate the STO of the address space to be traced.

**range**

restricts tracing to changes within the specified range.

The range can be a single address or a pair of addresses separated by a period (.) or a hyphen (-). A hyphen denotes a starting and ending range. A period denotes a starting address and a byte count. A data location is considered to be within the range if the first byte is within the range.

**hexloc**

restrict tracing to a specific location for the conditions set by the DATA and MASK operands. The operand *hexloc* can be any valid address. It cannot be a range of addresses.

**DATA hexstring**

specifies a hexadecimal data string to be matched when a store occurs into the specified *hexloc*. The length of the data string determines the bounds of the test. If the MASK option is also specified, the mask is logically applied to both storage and the DATA string before the comparison is made. The DATA operand requires the INTO *hexloc* operand.



**MASK hexstring**

specifies a hexadecimal string that forms a mask for data comparison when a store occurs in the specified *hexloc*. When you specify the MASK option, the mask is logically applied to both storage and the DATA string before the comparison is made. Only those bits and bytes that are binary ones are checked. The MASK operand requires the INTO *hexloc* operand and the DATA option. The length of the MASK string must be the same as the length of the string you specified for the DATA option.

**options**

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

**Usage Notes**

1. TRACE STORE might not trace storage alterations caused by the following instructions:
  - All DIAGNOSE instructions
  - All cryptographic facility instructions
  - Asynchronous I/O instructions
  - IUCV/VMCF/APPC
  - COMPRESSION CALL (CMPSC)
  - Monitor Call (MC)
  - Perform Frame Management Function (PFMF)
2. If the DATA option is specified, the length of the virtual storage that is monitored for alteration is defined by the length of the DATA string. If this length added to the specified hexloc generates a wraparound address range, the ending address of the storage to be monitored is set to the end of storage as defined by CP's current architecture.
3. Guest control register values and translation table contents define the address space structure of the virtual machine. Program execution (and the STORE command) can change that structure over time. The TRACE command operands that specify a guest address space to be traced by a trace trap use the guest control register and translation table values to determine the segment table origin (STO) of the address space to be traced at the time the TRACE command is issued. The STO is contained in the trace trap and can be examined by use of the QUERY TRACE command—that is, you specify an address space based on the state of your virtual machine at the time of the TRACE command that establishes the trace trap.
 

A trace trap generated with a specific address space specification (STO) always identifies that address space using the STO generated from the control registers and/or translation tables. If your program execution changes the control register values and translation table contents to represent a different address space structure, there is *no* change to the trace traps. The trace traps remain in effect with the address space specified at the time of the TRACE command execution.
4. When the operand *ALEThexword.ALraddr* is used, the primary list bit in the ALET is ignored. This bit chooses between access lists when the access list is not explicitly specified.
5. The operands PRI, SECO, HOME, *ASTERaddr*, *STDhexword*, *ASNasn*, *AREGareg*, *ALEThexword*, and *ALEThexword.ALraddr* use one or more of the access register translation (ART), ASN translation, and dynamic address translation (DAT) processes. These contain special case translation processes:
  - ALET X'00000000' always indicates the current primary address space designated by the STD in guest control register 1, regardless of the current contents of the dispatchable unit access list.
  - ALET 1 always indicates the current secondary address space designated by the STD in guest control register 7, regardless of the current contents of the dispatchable unit access list.
  - Access register 0 is always treated as containing ALET X'00000000'. The contents of access register 0 are not used during ART processing.
6. A traced address space is identified by its segment table origin (STO). The STD operand is provided for consistency with the DISPLAY, DUMP, and STORE commands. The STO is extracted from the STD provided, and all other STD fields are ignored.

7. If a guest specifies an address space option from which an ASN can be derived, then trace will trap events in all subspaces which are members of the group identified by the ASN whenever possible.

An ASN can be derived from the following address space options.

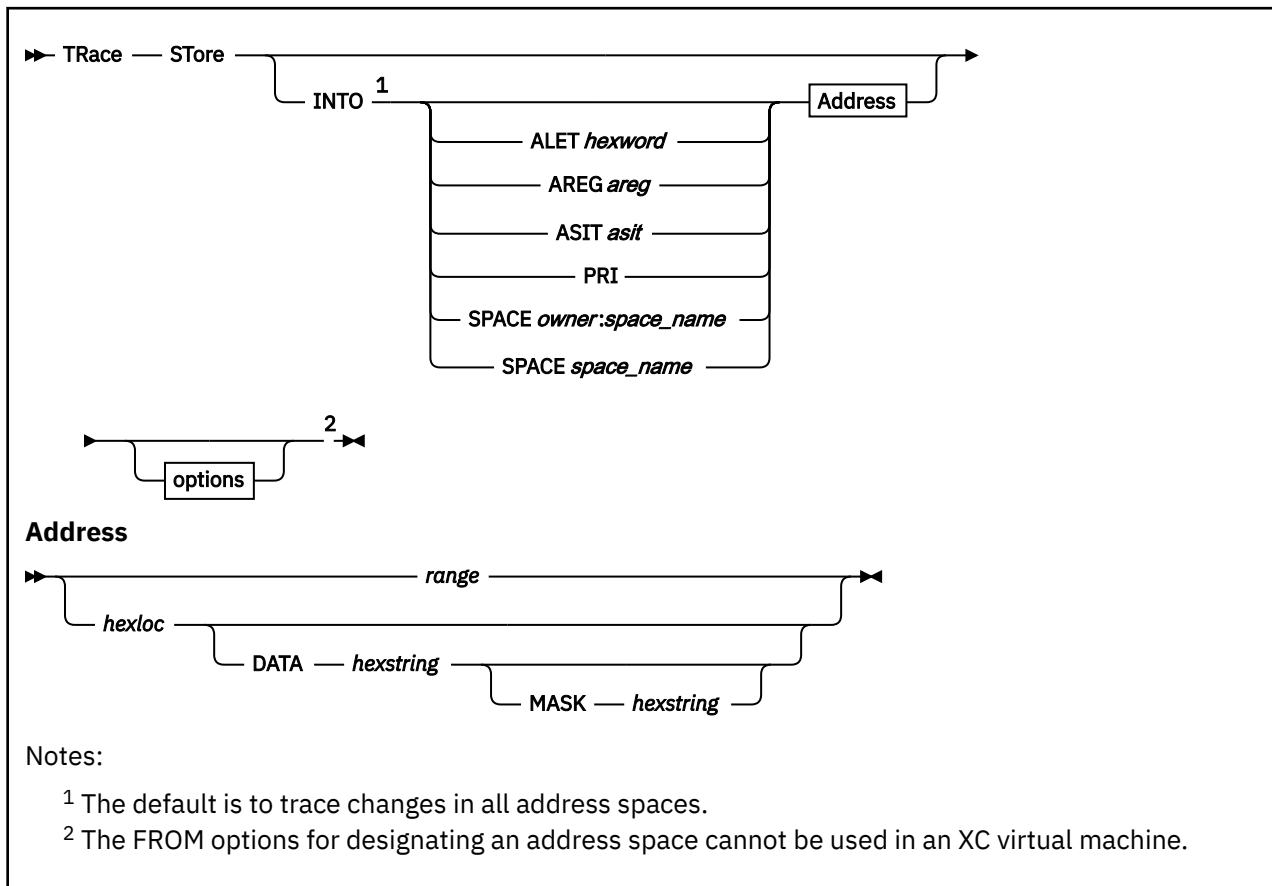
- PRI
- SECO
- ASN
- AREG0
- AREGx, where the ALET in ARx is 0 or 1
- ALETn, where n is 0 or 1

8. For more information, see TRACE [“Usage Notes”](#) on page 1946.

### Responses

For response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE STORE (ESA/XC, z/XC)



### Authorization

Privilege Class: G

### Purpose

Use TRACE STORE (ESA/XC, z/XC) to trace processor alterations made to virtual storage for an ESA/XC or z/XC virtual machine. You can trace all alterations, or only those alterations causing a specified set of logical locations to assume a specified value.

There is another version of the TRACE STORE command for ESA/390 and z/Architecture virtual machines. See [“TRACE STORE \(ESA/390, z/Architecture\)”](#) on page 1991.

### Operands

#### INTO

restricts tracing to a specific storage location or range of locations. Unless an address space is specified, changes to storage are traced in all address spaces.

#### ALET*hexword*

specifies that access register translation be used to locate the address space to be traced. The access list entry token (ALET) is used in conjunction with the host-access register translation tables to locate the address space to be traced.

#### AREG*areg*

specifies that access register translation be used to locate the address space to be traced. The access register (*areg*) is specified as one hexadecimal or two decimal digits. The access list entry token within

the access register is used in conjunction with the host-access register translation tables to locate the address space to be traced.

**ASIT***asit*

restricts tracing to events in the address space identified by the address space identification token, *asit*.

**PRI**

restricts tracing to the current primary address space.

**SPACEowner:space\_name**

limits tracing to the address space identified by *owner:space\_name*, where *owner*: specifies the user ID that owns the address space and *space\_name* is the name of the address space—the owner's user ID can be omitted if the owner is the user issuing the command. The space name is a string of alphanumeric and national characters, 1 to 24 characters long.

**range**

limits tracing to changes to at location within the specified range.

The range can be a single address or a pair of addresses separated by a period (.) or a hyphen (-). A hyphen denotes a starting and ending range. A period denotes a starting address and a byte count. A data location is considered to be within the range if the first byte is within the range.

**hexloc**

limits tracing to a specific location for the conditions set by the DATA and MASK operands. The operand *hexloc* can be any valid address. It cannot be a range of addresses.

**DATA hexstring**

specifies a hexadecimal data string to be matched when a store occurs into the specified *hexloc*. The length of the data string determines the bounds of the test. If the MASK option is also specified, the mask is logically applied to both storage and the DATA string before the comparison is made. The DATA operand requires the INTO *hexloc* operand.

**MASK hexstring**

specifies a hexadecimal string that forms a mask for data comparison when a store occurs in the specified *hexloc*. When you specify the MASK option, the mask is logically applied to both storage and the DATA string before the comparison is made. Only those bits and bytes that are binary ones are checked. The MASK operand requires the INTO *hexloc* operand and the DATA option. The length of the MASK string must be the same as the length of the string you specified for the DATA option.

**options**

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

## Usage Notes

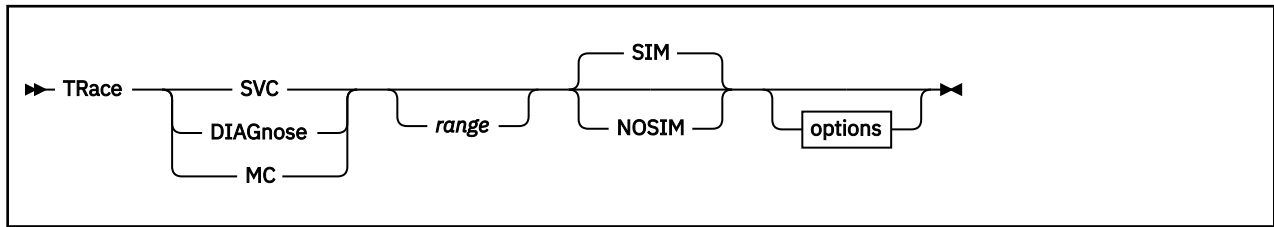
1. TRACE STORE might not trace storage alterations caused by the following instructions:
  - All DIAGNOSE instructions
  - All cryptographic facility instructions
  - Asynchronous I/O instructions
  - IUCV/VMCF/APPC
  - COMPRESSION CALL (CMPSC)
  - Monitor Call (MC)
  - Perform Frame Management Function (PFMF)
2. If the DATA option is specified, the length of the virtual storage that is monitored for alteration is defined by the length of the DATA string. If this length added to the specified hexloc generates a wraparound address range, the ending address of the storage to be monitored is set to the end of storage as defined by CP's current architecture.
3. The operands PRI, AREG*areg*, ALET*hexword* use the host-access register translation (ART) process, which includes these special cases:
  - ALET X'00000000' always indicates the current host-primary address space.

- Access register 0 is always treated as containing ALET X'00000000'. The contents of access register 0 are not used during ART processing.
4. TRACE STORE traps that are set for a specific address space while the virtual machine is using the XC architecture do not cause instructions to be traced when the architecture is changed to ESA. In contrast, traps that apply to all address spaces remain effective when the virtual machine architecture is changed and function as though they had been created in the new architecture.
  5. For more information, see TRACE [“Usage Notes”](#) on page 1946.

## **Responses**

For response information, see [“TRACE: Command Responses”](#) on page 1954.

## TRACE SVC/DIAGNOSE/MC



### Authorization

Privilege Class: G

### Purpose

Use TRACE SVC, TRACE DIAGNOSE, and TRACE MC to trace the following:

- Supervisor calls (SVC)
- DIAGNOSE instructions
- Monitor calls (MC).

**Note:** Tracing is limited to those instructions that cannot result in an operation or privileged operation exception.

### Operands

#### *range*

identifies a hexadecimal range of SVC numbers, DIAGNOSE codes, or monitor call codes. A range may be specified as a single number or pair of numbers separated by a period (.) or a hyphen (-). Wrap-around ranges are not allowed.

#### **SIM**

specifies normal simulation of instructions. If neither SIM nor NOSIM is specified, SIM is assumed.

#### **NOSIM**

specifies suppression of instruction simulation. Program interrupts for guest PER events are also suppressed. By using NOSIM and the CMD option, you can define a CP command or set of CP commands to be processed in place of the suppressed instruction. The PSW is left pointing to the next sequential instruction.

#### *options*

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

### Usage Notes

1. SVC, DIAGNOSE, and MC instructions that cause an operation or privileged operation exception are not displayed.
2. TRACE MC will not provide output for the Monitor Call instructions when the enhanced-monitor facility is used and a monitor-event-counting operation is performed.
3. For general usage information for TRACE commands, see TRACE [“Usage Notes” on page 1946](#).

### Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE TABLE

► TRace — TAbLe ◄

### Authorization

Privilege Class: G

### Purpose

Use TRACE TABLE to display the traceback table.

### Usage Notes

1. The traceback table is a list of successful branch instructions that is maintained while branch tracing is active. It contains the most recent branch instructions processed. If the same instruction is processed with no intervening branch instructions, the number of times the instruction was processed will be maintained.

Up to six instructions are maintained in the traceback table. The most recent branch is displayed as the last entry in the table.

The contents of the breaking-event-address register are displayed after the traceback table if the PER-3 facility is available.

2. For more information, see TRACE [“Usage Notes”](#) on page 1946.

### Responses

Response 1:

```
INST-ADDR INST-STO MNEM  OP-OPER   DEST-ADR DEST-STO COUNT
aaaaaaaa ssssssss mnem xxxxxxxxxxxx -> tttttttt ssssssss nn
```

Last Breaking-Event-Address = aaaaaaaaa

is displayed if there is at least one entry in the traceback table and all entries in the table were collected in ESA/390 mode. This may be repeated for up to six entries. The operand *nn* specifies the number of times in a row this branch instruction was processed. This value is not displayed if *n*=1.

The STO fields are filled with blanks when branch instructions are running in the real address mode.

Response 2:

```
INST-ADDR      MNEM  OP-OPER   DEST-ADDR      COUNT
aaaaaaaa_aaaaaa mnem xxxxxxxxxxxx -> tttttttt_tttttttt nnnnnnnnn
ASCE=ssssssssssssssssssss ASCE=ssssssssssssssssssss
```

Last Breaking-Event-Address = aaaaaaaaa\_aaaaaa

is displayed if there is at least one entry in the traceback table and at least one entry in the table was collected in z/Architecture mode. This may be repeated for up to six entries. The operand *nn* specifies the number of times in a row this branch instruction was processed. This value is not displayed if *n*=1.

The STO and ASCE fields are filled with blanks when branch instructions are running in the real address mode.

The response below shows output with two ESA/390 and three z/Architecture table entries:

```
INST-ADDR      MNEM  OP-OPER   DEST-ADDR      COUNT
aaaaaaaa_aaaaaa mnem xxxxxxxxxxxx -> tttttttt_tttttttt nnnnnnnnn
```

## TRACE TABLE

```
STO =ssssssss          STO =ssssssss
aaaaaaaa_aaaaaaaaa mnem xxxxxxxxxxxxxxxx -> tttttttt_tttttttt nnnnnnnnnn
STO =ssssssss          STO =ssssssss
aaaaaaaa_aaaaaaaaa mnem xxxxxxxxxxxxxxxx -> tttttttt_tttttttt nnnnnnnnnn
ASCE=ssssssssssssssss ASCE=ssssssssssssssss
aaaaaaaa_aaaaaaaaa mnem xxxxxxxxxxxxxxxx -> tttttttt_tttttttt nnnnnnnnnn
ASCE=ssssssssssssssss ASCE=ssssssssssssssss
aaaaaaaa_aaaaaaaaa mnem xxxxxxxxxxxxxxxx -> tttttttt_tttttttt nnnnnnnnnn
ASCE=ssssssssssssssss ASCE=ssssssssssssssss
Last Breaking-Event-Address = aaaaaaaaa_aaaaaaaaa
```

Entries collected in z/Architecture mode are formatted with ASCE=ssssssssssssssss; entries collected in ESA/390 mode are formatted with STO=ssssssss.

Whether or not an underscore is used in the address fields depends on the SET UNDERSCORE setting.

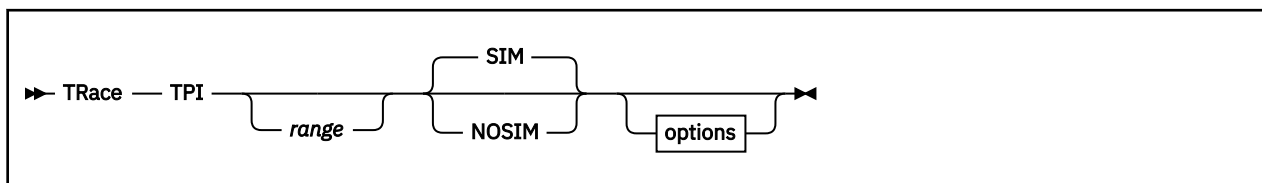
Response 3:

```
Traceback table is empty
```

is returned when there are no entries in the traceback table.



## TRACE TPI



### Authorization

Privilege Class: G

### Purpose

Use TRACE TPI to trace test pending interruption instructions.

**Note:** TRACE TPI is accepted if your virtual machine is an ESA or XC virtual machine; tracing is limited to those instructions that do not produce an operation or privileged-operation exception.

### Operands

#### *range*

is a range of device numbers. If you specify a device range, only those TPI instructions that store an interruption is traced. If you do not specify a range, all TPI instructions are traced. The device range applies to the device name, and not the subchannel address.

The range may be a single device number, a device number followed by a period and followed by a number that represents the number of consecutive device addresses to be traced, or pair of device numbers separated by a period (.) or a hyphen (-). Wrap-around ranges are not allowed.

NOSIM and a device range are not compatible options.

#### **SIM**

specifies normal simulation of instructions. If neither SIM nor NOSIM is specified, SIM is assumed.

#### **NOSIM**

specifies suppression of instruction simulation. Program interrupts for guest PER events are also suppressed. By using NOSIM and the CMD option, you can define a CP command or set of CP commands to be processed in place of the suppressed instruction. The PSW is left pointing to the next sequential instruction.

This operand is not valid when TPI specifies a device range.

#### *options*

For an explanation of the options, see [“TRACE: Options” on page 1949](#).

### Usage Notes

1. For usage information, see TRACE [“Usage Notes” on page 1946](#).

### Responses

For response information, see [“TRACE: Command Responses” on page 1954](#).

## TRACE TRAP



### Authorization

Privilege Class: G

### Purpose

Use TRACE TRAP to alter a named trace trap.

### Operands

#### TRAP

applies the options you specify to a single event.

All primary TRACE operands establish units of control called traps. Each trap, except for TRACE ALL, is assigned a unique identifier and has its own set of options, such as TERM, PRINT, RUN, and STEP.

TRACE ALL creates a unique set of traps replacing any active TRACE ALL command. These traps each have an identifier of "ALL," which may be used for changing trace options or deleting the trap. For those traps for which SIM and NOSIM are valid options, you can specify SIM or NOSIM with TRAP.

Any valid option you may specify with the TRAP operand you may specify without any primary TRACE operand. Not specifying a primary TRACE operand has the same effect as a TRACE TRAP command processed for each event in the current trace set.

#### *ident*

identifies a trace trap describing the event; it must be one to four characters long. If a trap already exists with the given identifier, it is replaced by the trap for the newly defined event. You cannot use the word ALL as an identifier; on the TRACE TRAP command, this option is not valid.

#### *options*

except for ID, you can specify any of the options. For a full explanation of the options, see ["TRACE: Options" on page 1949](#).

### Usage Notes

1. Note that you cannot specify the ID option with TRACE TRAP to change the identifier of an existing trap. In order to change the identifier, you must redefine the trap.
2. With the exception of SIM and NOSIM, you can use TRACE TRAP to modify *only* those options listed here. To change unique options not listed here, you must redefine the trap. However, you can use TRACE TRAP to modify the SIM and NOSIM on any trap for which they are valid options.
3. For more information, see TRACE ["Usage Notes" on page 1946](#).

### Responses

For response information, see ["TRACE: Command Responses" on page 1954](#).

## TRACE TXSUSPEND/NOTXSUSPEND



### Authorization

Privilege Class: G

### Purpose

Use TRACE TXSUSPEND and TRACE NOTXSUSPEND to suppress (or to not suppress) trace events during transactional execution.

### Operands

#### TXSUSPend

marks the active trace set so that tracing will be suspended while the virtual CPU is in transactional-execution mode. This allows guest transactions to complete without interference from TRACE. TRACE does not recognize, report or act on any events that occur within a transaction.

#### NOTXSUSPend

marks the active trace set so that tracing will remain active during transactional execution. This ensures that such events will not occur unreported. However, this setting can cause transactions to be persistently aborted. See [“Usage Notes” on page 2003](#).

### Usage Notes

1. TXSUSPEND and NOTXSUSPEND apply per trace set but can only be altered for the active trace set. The initial setting when a trace set is created is NOTXSUSPEND. This setting affects trace traps of types INSTRUCTION, STORE, BRANCH, G, GG, and AR.
2. Trace traps of type INSTRUCTION, STORE, BRANCH, G, GG or AR can interfere with transactional execution (for example, with using the TBEGIN instruction) in the virtual machine. If these traps cause an event to be recognized in transactional-execution mode, then the transaction is aborted, message HCP2106I is displayed and the virtual machine is placed in CP READ.

This interference can occur even for events that do not match the criteria specified in the trace traps. For example, address ranges specified on multiple traps can be merged, resulting in recognition of events in the gaps. Likewise, tracing register alterations triggers an event after every instruction, to determine whether the specified registers have changed.

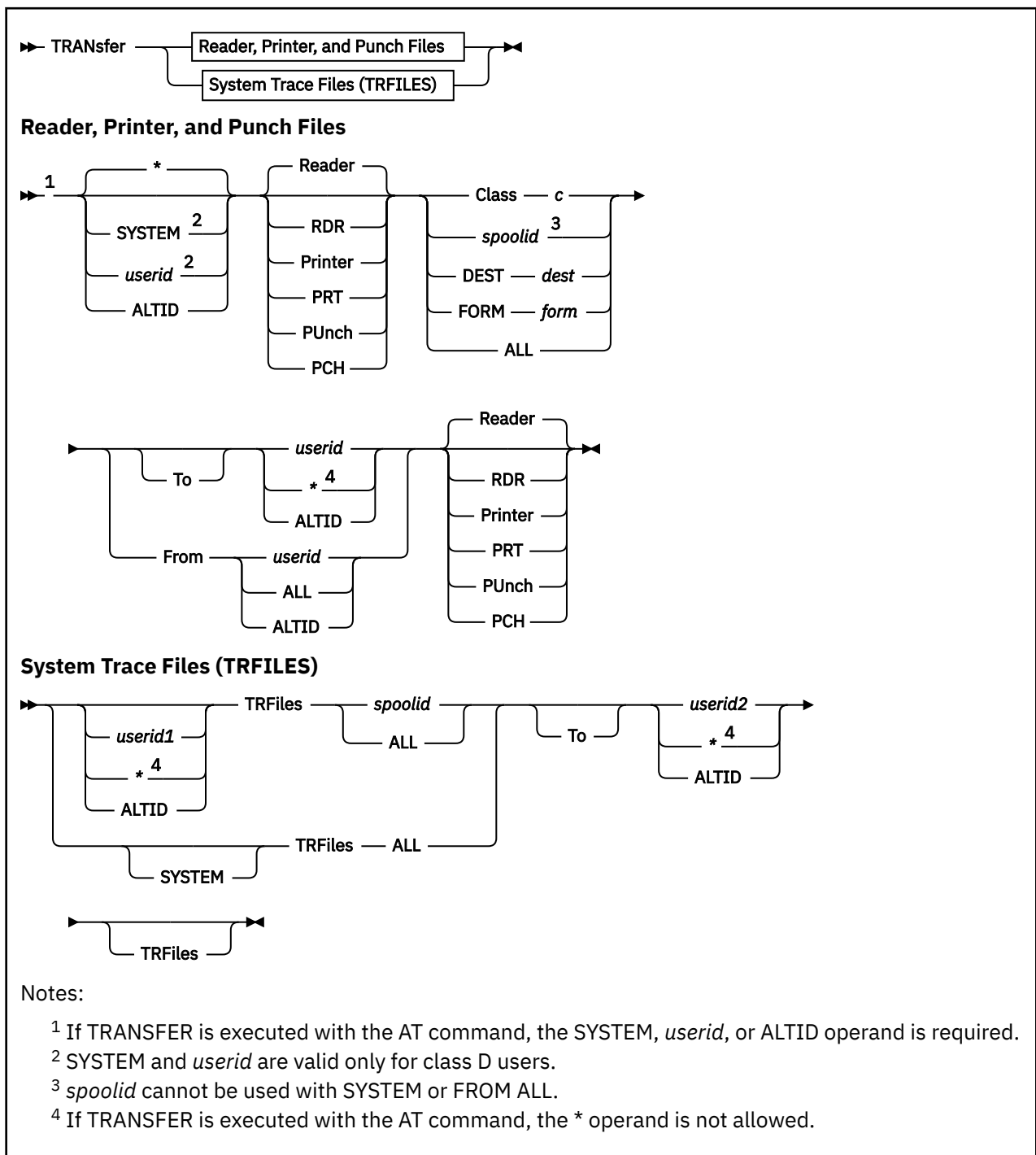
Without additional intervention following the HCP2106I message, the virtual machine will next run from the transaction abort instruction address. This might result in the transaction being retried and a perpetual cycle of trace events to occur causing the transaction to abort. The user has the option to suppress trace events whenever the virtual machine is running in transactional execution mode by issuing TRACE TXSUSPEND.

3. TRACE (NO)TXSUSPEND uses the Program Event Recording (PER) event-suppression control in the machine. If the guest program is also using PER, the TRACE setting for this control must agree with the guest's use. Otherwise, message HCP2107E is generated, and the virtual machine can not be resumed until the conflict is resolved, for example by changing the TXSUSPEND setting or switching to a null trace set.
4. For more information, see TRACE [“Usage Notes” on page 1946](#).

**Responses**

For response information, see [“TRACE: Command Responses”](#) on page 1954.

# TRANSFER



## Authorization

Privilege Class: D, G

## Purpose

## TRANSFER

Use TRANSFER to send files to, or retrieve them from, the reader, printer, or punch queue of another user. The TRANSFER command may also be used to send system trace files (TRFILES) to another user's TRFILE queue.

Class G users can manipulate files that they originate or own on their primary or alternate user IDs. Class D users can transfer any spool files in the system, regardless of ownership or origin.

If you do not specify any of the options (SYSTEM, *userid*, or asterisk [\*]), the default (\*) is to transfer or reclaim your own spool files.



**Attention:** You should not attempt to use this command to transfer a store-and-forward file away from RSCS and then back to it. The file might be lost.

### Operands

#### SYSTEM

allows the Class D user to manipulate all spool files in the system, regardless of their ownership. This is not allowed with the *spoolid* option.

#### *userid*

\*

#### ALTID

identifies the user whose spool files are to be transferred or reclaimed. If you are manipulating your own files, you may specify *userid* as an asterisk (\*). This is the default. You may specify ALTID to manipulate files owned by your alternate user ID.

The *userid* option allows the Class D user to manipulate files belonging to any user on the system. Class G users can use the *userid* option to manipulate files that they currently own or have originated.

#### Reader

#### RDR

#### Printer

#### PRT

#### PUnch

#### PCH

specifies the queues to which CP transfers files and from which files are reclaimed. READER is the default.

#### Class *c*

is the class of spool files you wish to transfer. The operand *c* is a 1-character alphanumeric field with values from A to Z and from 0 to 9.

#### FORM *form*

selects spool files with the specified form for transferring. For the Class G user, the *form* is a 1- to 8-character form name. For the Class D user, *form* is a form name when you specify asterisk (\*) or default to your user ID. Otherwise, *form* is the operator form number.

#### DEST *dest*

selects spool files with the specified destination for transferring. The field *dest* is a 1- to 8-character name.

#### ALL

specifies a transfer of all files in a reader/printer/punch queue.

#### *spoolid*

identifies the file that you want to transfer or reclaim. This operand is not valid with the SYSTEM or FROM ALL keywords.

#### TRFiles

specifies that the spool file is coming from or going to the system trace file queue.

## Options

### TO *userid*

### TO \*

### TO ALTID

identifies the user to whom you want to transfer the files. If you omit the optional keyword TO, *userid* cannot be TO or T. You can use an asterisk (\*) to denote your own user ID. ALTID indicates that you want to transfer files to your alternate user ID.

### FROM *userid*

### FROM ALL

### FROM ALTID

identifies the user from whom you want to reclaim spool files of the specified reader, printer, or punch queue. You can specify ALL to reclaim all files that you originally created and transferred to the reader/printer/punch queues of other users. You can specify ALTID to get files that belong to your alternate user ID.

## Usage Notes

1. SET IMSG OFF suppresses informational messages for this command.
2. You cannot use the TRANSFER command to transfer any active spool files.
3. If the NORETURN option is specified when the file is created by DIAGNOSE X'94', the originator of the file cannot retrieve that file from the current owner. If you try, you receive the following message:

```
FILE NOT TRANSFERRED;
FILE CREATED WITH THE NORETURN OPTION
```

4. When a file changes ownership, as when it is changed to or from another user, the spool file ID number also changes. The new number is obtained from the pool of available IDs of the recipient of the file. Thus, if user BINGLE changes file number 3 to user BOGUS, BINGLE gets the response:

```
RDR FILE 0003 SENT TO BOGUS RDR AS 0018
RECS 1000 COPY 001 A NOHOLD NOKEEP
```

BOGUS now has a file number 18.

5. If a converted file (a file that has been preprocessed by the print server) is transferred to a new user, the old file descriptor is kept. A new file is created for the new owner, but the old file descriptor remains until the print server acknowledges that the old file is no longer wanted.
6. If DEST is not specified on a command that can select files by DEST, the destination value of the file is ignored.
7. System trace files (TRFILES) may only be transferred from one user's TRFILES queue to another user's TRFILES queue. Only system trace files may be transferred to a TRFILES queue.
8. System trace files (TRFILES) cannot be retrieved from another user with the FROM option of the TRANSFER command. To retrieve TRFILES, either the new file owner or a Class D user must use the TO option of the TRANSFER command.
9. If an external security manager is installed on your system, you may not be authorized to enter this command. For additional information, contact your security administrator.
10. In an SSI cluster, no files are transferred if all of the following conditions are true:
  - The command issuer is defined as a multiconfiguration virtual machine.
  - The file being transferred is a copy of a file that was created on another member of the SSI cluster.
  - The command issuer is the user to whom the file is being transferred.

## Examples

```
transfer 0123 to usera
```

## TRANSFER

transfers the spool file with spoolid 0123 from your reader queue to the reader queue of USERA. Note that the file is in the reader queue, which is the default. If you want to transfer a printer or punch file, you must specify the device type in the command line.

```
transfer prt 0123 to usera
```

transfers the spool file with spoolid 0123 from your print queue to the reader queue of USERA.

```
transfer prt all to * rdr
```

transfers all the spool files on your print queue to your reader queue. Note that you do not need to specify the default, RDR.

```
transfer rdr class b to * pch
```

transfers all the Class B files on your reader queue to your own punch queue.

```
transfer prt all from all rdr
```

reclaims all spool files that you sent to the printer queues of other users and places them on your reader queue.

## Responses

Response 1:

Both the current and the previous owners of the file receive the following response when the previous owner issues a TRANSFER command with the TO option. It is also the response that the previous owner of the file receives if the previous owner issues a TRANSFER command with the FROM option.

**Note:** In the response sent to the user ID receiving the file, *spd1* and *spd2* will be in the reverse order of that shown in the following response.

```
{RDR} FILE spd1 SENT {TO } userid {RDR} {AS } spd2 RECS {nnnn}  
{PRT}                {FROM}      {PRT} {WAS}      {nnnK}  
{PUN}                {PUN}      {PUN}      {nnnM}
```

Continued:

```
CPY [*] nnn c {HOLD } {KEEP }  
              {NOHOLD} {NOKEEP}
```

or, for system trace files (TRFILES):

```
TRF FILE spd1 SENT {TO } userid TRF AS spd2 RECS nnnn  
                  {FROM}      nnnK  
                          nnnM
```

### RDR PRT PUN TRF

(first occurrence). If you are transferring a file to another user's virtual machine, this field indicates the queue in which the file originated. If you are receiving a file from another user, this field indicates the file queue from which the file has been added.

### **spd1**

identifies the file at the time the command is issued.

### **TO userid**

identifies the new file owner.

### **FROM userid**

identifies the previous file owner.



**RDR**  
**PRT**  
**PUN**  
**TRF**

(second occurrence). If you are transferring a file to another user's virtual machine, this field indicates the queue to which the file has been added. If you are receiving a file from another user, this field indicates the file queue in which the file originated.

**spd2**

identifies the file when the command is completed.

**RECS *nnnn***

**RECS *nnnK***

**RECS *nnnM***

identifies the number of logical records in the spool file. If the number is greater than 9999, the number is shown as *nnnK*, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, the number is shown as *nnnM*, where M represents the number of lines rounded to the nearest million.

**CPY [\*] *nnn***

is the remaining number of copies of the file to be produced. The asterisk (\*) is for the CP-driven 3800 printer and allows the printer to copy each page *nnn* times before going on to the next page.

**c**

is a 1 character alphanumeric field representing the class with a value from A to Z and from 0 to 9.

**HOLD**

**NOHOLD**

is the user HOLD status associated with the file.

**KEEP**

**NOKEEP**

is the KEEP status associated with the file.

Response 2:

You receive the following response when a converted file is transferred to a new owner, or to the reader or punch queue. CP assigns a new *spoolid* to the file and marks it as unconverted.

```
SPOOLID spd1 HAS BEEN CHANGED TO spd2
```

**spd1**

identifies the file before it is transferred.

**spd2**

identifies the file after it is transferred.

Response 3:

```
{nnnnnnn} FILE(S) TRANSFERRED
{          NO}
```

identifies the number of files transferred. NO indicates that there are no files matching the characteristics (such as CLASS and FORM) specified on the TRANSFER command.

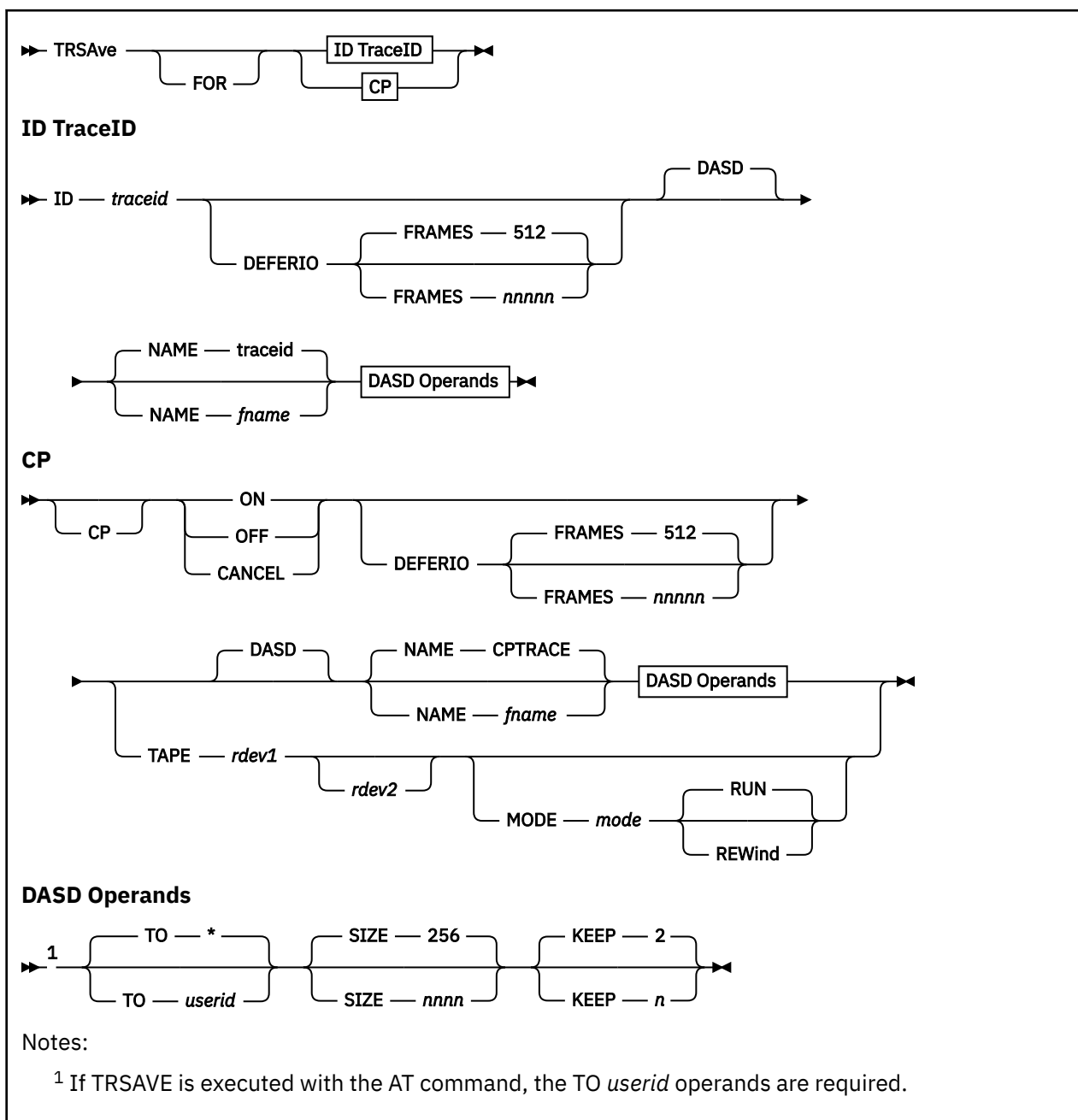
## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*
- HCP006E Invalid device type - *{rdev/vdev/ldev}*
- HCP007E Invalid userid - *userid*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid

## TRANSFER

- HCP028E Class missing or invalid
- HCP035E Device type missing or invalid
- HCP042E User *userid* spoolid *nnnn* does not exist
- HCP260E {Command|*command* command} processing cannot complete due to errors.
- HCP356E Access denied; User *userid* file *spoolid* not {changed|transferred|printed}
- HCP364E No alternate userid is currently set.
- HCP413E Command processing cannot complete due to a spool paging error.
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP429E *type vdev* spool error; file {held| purged}
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP644E Unknown/unauthorized CP command: *command*
- HCP848E SYSTEM parameter is invalid with the spoolid option
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option* - *operand*.
- HCP1552E More than {4|8} operands were specified for the {CHARS|CLASS} option.
- HCP1560E File was not {changed|transferred} because it was created with the NORETURN option.
- HCP2505E An output device is an invalid destination for system data files.
- HCP6525E External Security Manager is unavailable.

# TRSAVE



## Authorization

Privilege Class: A, C

## Purpose

Use TRSAVE to specify where the following data is to be saved:

- CP trace table data
- Data from traces defined by TRSOURCE.

**Note:** CP trace table data may be saved in system trace files (TRFILEs) or on tape. Data from traces defined by TRSOURCE may be saved only in system trace files.

For more information, see [“QUERY TRFILES” on page 1231](#) and [z/VM: Diagnosis Guide](#).

## Operands

### ID *traceid*

indicates that this TRSAVE applies to data generated by *traceid*, which was defined by the TRSOURCE command.

### CP ON

indicates that the recording of CP trace table data should begin.

### CP OFF

stops CP system trace table data recording. Any remaining buffers of data are written to the output medium. Either a tape mark is written, or the DASD file is closed, depending on the output medium. Tapes are rewound and unloaded.

### CP CANCEL

stops CP system trace table data recording. Any pending I/O is canceled. Any data collected, but not written, is lost. When tape is the output medium, a tape mark is written, and the tape is rewound and unloaded, regardless of the previous tape positioning option.

### DEFERIO

indicates that no trace data is to be recorded on DASD or TAPE until the *traceid* is disabled or, for CP trace table recording, until TRSAVE CP OFF is issued. Until that time, when all in-storage buffers are filled tracing wraps to the oldest buffer. DEFERIO is not valid for TYPE GT BLOCK mode traces.

### FRAMES *nnnnn*

specifies the number of real storage frames to be used for the in-storage-wrap buffers. The default is 512 frames. The minimum is 8 frames. The maximum is 99999 frames per trace, but may be further limited by the number of pageable pages in the system dynamic paging area that are eligible for DEFERIO traces. See usage note [“17” on page 2014](#) for usage notes regarding DEFERIO and FRAMES.

### DASD

indicates that trace data is to be recorded in system trace files. DASD is the default.

### TO *userid*

\*

The operand *userid* specifies the user ID to receive the system trace file containing the trace data.

The operand asterisk (\*) specifies that the user who issues the command should receive the system trace file containing the trace data. The operand asterisk (\*) is the default.

### NAME *fname*

specifies the file name to be assigned to the system trace file. For CP system trace table data, CPTRACE is the default name. For trace data generated by traces defined by the TRSOURCE command, the trace ID is the default name.

### SIZE *nnnn*

indicates the maximum size of the system trace files. The operand *nnnn* is a decimal number that specifies 4 KB pages of trace data. Once the file reaches the limit, it is closed and a new file is opened when the next trace record is received. The default is 256 pages. The minimum is eight pages. The maximum is 9999 pages.

### KEEP *n*

specifies the total number of trace files to be kept on DASD for this recording function. The default is two files. The maximum is nine files.

### TAPE

indicates that trace data is to be recorded on tape. This option is valid only for recording CP system trace tables.

### *rdev1* [*rdev2*]

indicates that the CP trace table data captured by this trace should be recorded on the tape drive at the specified device number. Either one or two tape drives can be specified on any tape option. These tape drives must be system drives and must not be attached to any users.

If two tape drives are specified, CP automatically switches to the alternate tape drive when it encounters an end-of-tape mark.

### MODE mode

selects the recording format in which data will be written on the tape. On 3422 tape devices, you can specify the recording format (the MODE) only at the beginning of a tape volume. No changes in the format can be made after this initial designation. On 3480, 3490 (including 3490Es), and 3590 tape devices, you can change the recording format (the MODE) each time you choose to write to a tape volume. After you write to a volume, the MODE value returns to the default value for that particular tape device. If you wish to write to the tape in a different recording format next time, you must again specify the MODE value.

The table below specifies the valid and default mode option values for supported tape devices:

Device Type	Valid Device-Dependent (MODE mode) Option Values	Valid Device-Independent (MODE nnnn) Option Values	Device (MODE mode) Default Value
3422	1600, 6250	n/a	1600
3480	38K	n/a	38K
3480 w/IDRC	XF, 38K	COMP	38K
3490	3490C	COMP	3490C
3590	n/a	COMP	COMP

To determine which device to use, attach the tape device to your *userid* and issue the 'Q V TAPE' command. Then detach the tape device.

### RUN

specifies that the tape is rewound and unloaded when it fills. This is the default.

### REWIND

specifies that the tape is rewound but not unloaded when it fills. Recording continues to either the other tape drive devoted to CP tracing or the tape drive on which the tape was just filled.

## Usage Notes

1. The destinations for trace data are not saved across system hard abends. Therefore, the destination for the trace data should be saved in EXECs.
2. The parameter TAPE and tape options can only be used on TRSAVE FOR CP commands.
3. TRSAVE FOR ID *traceid* cannot be issued when *traceid* has been enabled using the TRSOURCE command.
4. The optional destination parameters, TO, NAME, SIZE, and KEEP for DASD, and MODE, RUN, REWIND for TAPE may be specified in any order.
5. For TRSAVE FOR ID *traceid*, the specified parameters remain in effect until the trace ID is dropped or they are changed on a subsequent TRSAVE command. For example, if you enter the command `trsave id debug keep 5 size 9`, and then enter `trsave id debug size 20`, the *traceid* DEBUG has TRSAVE options *traceid* recorded to system trace files with a file name of DEBUG, a size of 20 pages, a keep maximum of five files, and a receiver of your user ID.
6. The use of two tape drives is recommended to minimize loss of data. With one tape drive, data may be lost while you rewind and reset for the next tape. If two tape drives are used, both tape drives must support the specified MODE.

The use of the REWIND option is also recommended. If the REWIND option is not used, a full tape is rewound and unloaded. If one tape drive is used without the REWIND option, data may be lost until the operator mounts another tape. Even if two tape drives are used, data may be lost on heavily loaded systems when the second tape drive becomes full if the operator has not yet:

- Mounted another tape on the first tape drive
  - Completed the operation intervention on the first drive.
7. Table 48 on page 2014 shows the default file names for system trace files and their file types. The file names may be changed with the NAME *fname* option.

Table 48. Default File Names and File Types for System Trace Files

Source of Trace	File Name	File Type
CP system trace data	CPTRACE	CP
TRSOURCE ID <i>traceid</i> TYPE DATA	<i>traceid</i>	DATA
TRSOURCE ID <i>traceid</i> TYPE IO	<i>traceid</i>	IO
TRSOURCE ID <i>traceid</i> TYPE GT FOR USER	<i>traceid</i>	VM
TRSOURCE ID <i>traceid</i> TYPE GT FOR VMGROUP	<i>traceid</i>	VMG

8. When a trace is active, the files associated with it have a class of W. After tracing is complete, the associated files have a class of A.
9. On stressed systems, the trace table may wrap before all entries are saved. In this case, a special trace entry is made to denote data loss.
10. TRSAVE CANCEL immediately terminates the TRSAVE function. Any I/O that is pending is canceled, and any information that has been merged and not written is lost. To ensure that all information collected has been written, use the OFF option on the TRSAVE command.
11. TRSAVE ON is valid only when TRSAVE is not currently active. Likewise, TRSAVE OFF is valid only when TRSAVE is currently active. You may use TRSAVE CANCEL while TRSAVE OFF is in process (for example, when the tape is in a state that requires intervention), but it terminates any I/O that is currently pending.
12. The system attempts to write a tape mark at the end of the data on tape when OFF and CANCEL processing is completed. This attempt may fail because of irrecoverable I/O errors on the tape, but you may still process the tape using the TRACERED utility.
13. KEEP and SIZE may be used together to manage the system spool space. They limit the total amount of space available to record trace data for this trace request. CP limits the amount of data in each file, according to the SIZE specified. By designating a number of files to be kept, TRSAVE allows CP to purge the oldest trace file, keeping the most recent data and continuing to trace when the allowable space limit is reached.
14. Each time a trace becomes active, the trace data is recorded in another file or set of files. For example, if you enter the following commands, you can generate up to six files:

```
trsave for cp dasd keep 3
trsave off
trsave for cp dasd keep 3
trsave off
```

15. For 3490 tape devices, models A01, A02, B02, B04, D31, and D32 should be treated as type 3480 devices. All other 3490 device models should be treated as type 3490 devices. For a complete breakdown of 3480 and 3490 device types by model number, see [Defining I/O Devices in z/VM: CP Planning and Administration](#).
16. If an external security manager is installed on your system, you may not be authorized to enter the TRSAVE command with the TO operand. For additional information, contact your security administrator.
17. Usage notes regarding DEFERIO
- a. The frames are allocated from the system's dynamic paging area when the traceid is enabled or, for CP trace table data, when TRSAVE CP ON DEFERIO FRAMES *nnnnn* is issued. System performance could be impacted because of the frames not being available for system dynamic paging.

- b. A certain amount of the pageable (non-fixed) pages in the system dynamic paging area are eligible to be allocated for DEFERIO traces. That amount is a percentage (75%) of the total amount of pageable pages in the system dynamic paging area.

A disabled DEFERIO traceid could have the entire amount of eligible pages in the system dynamic paging area specified regardless of the amount of frames specified for other DEFERIO traces. In order to enable the traceid however, no more than the entire amount of eligible pageable pages in the system dynamic paging area can be allocated for the sum of all active DEFERIO traces.

If an attempt is made to enable a DEFERIO traceid that could result in more than the eligible amount of pageable pages in the system dynamic paging area being allocated to active DEFERIO traces, then that TRSOURCE ENABLE command would fail.

Use QUERY TRSAVE FRAMES command to determine how many frames have been allocated from the system dynamic paging area for active DEFERIO traces and to determine how many pageable pages in the system dynamic paging area remain eligible to be used for DEFERIO traces.

Keep in mind that the number of pageable pages in the system dynamic paging area is affected by the locking and unlocking of pages for various reasons, so the amount of pages eligible to be allocated to DEFERIO traces is a fluctuating amount. You should avoid specifying a FRAMES value that is borderline to the total amount of eligible pages. If the total amount of eligible pages were to decrease after the TRSAVE command has accepted the frames value and before you enable the traceid, then the TRSOURCE ENABLE command could fail because of the inability to allocate the frames at that time.

- c. For every 990 frames, or up to 990 frames allocated for a DEFERIO trace, CP locks an additional frame in the dynamic paging area for its use.
- d. If a large number of frames are specified, then it may take some time to steal those frames. The TRSOURCE ENABLE command will not complete (and CP trace table recording will not be turned on) until the specified amount of frames is allocated. When the number of FRAMES specified on the TRSAVE command for a DEFERIO traceid trace is greater than 512, informational messages will be issued to document the progress of obtaining the frames.

If a large number of frames are specified, then it may take some time for the TRSOURCE DISABLE command to complete (and for CP trace table recording data to be written) because at that time all of the frames must be written to DASD or to TAPE. When the number of frames to be written out for a DEFERIO traceid trace is greater than 3, informational messages will be issued to document the progress of writing the frames.

Because the issuing virtual machine will be in 'command wait' until the command completes, you should not enter the TRSOURCE ENABLE or TRSOURCE DISABLE commands from a virtual machine that may be adversely affected by command wait, such as a server virtual machine, a secondary virtual machine, or a PROP virtual machine.

- e. When a DEFERIO trace is eventually recorded on DASD, the SIZE and KEEP operands affect the size and number of trace files that are kept. You should specify a combination of SIZE and KEEP that is equivalent to the number specified for FRAMES. That is,  $SIZE * KEEP = FRAMES$ .
- f. If the system terminates with a DEFERIO trace active, then no data is recorded on DASD or TAPE for that trace. However, the trace data can be extracted from a CP dump.

### Examples

- To record system trace table entries on DASD and specify USERA as the file owner, enter **trsave for cp on dasd to usera**.
- To turn off TRSAVE for CP, enter **trsave for cp off**.
- To turn on TRSAVE for the trace defined by TRSOURCE ID DEBUG, save the recorded data in a file named FILEB, and allow DUSER to process the files, enter **trsave for id debug dasd to duser name fileb**.
- To use in-storage wrap for the same traceid so that data is not written to DASD until the traceid is disabled, enter **trsave for id debug deferio frames 1000 size 500 keep 2**.

## Responses

Response 1:

For CP tracing:

```
TRSAVE function { started }
                { completed }[: nnnn files purged]
                { canceled }[: nnnn files purged]
```

### **nnnn**

is the number of trace files that have been purged during the time the trace service tool function was active because of the wrapping of the keep limit. When the keep limit is reached, the oldest file is purged. If the response states that 9999 files were purged, it is possible that more than 9999 files were purged.

You receive this when you enter `trsave for cp` and the TRSAVE command is started, completed, or canceled. If you are not logged on, the completed and canceled responses are sent to the system operator.

No response is issued when TRSAVE FOR ID *traceid* commands complete because, while TRSAVE defines the destination of traces defined by TRSOURCE, recording of those traces is started and stopped by the ENABLE and DISABLE parameter of the TRSOURCE command.

With both the TRSAVE `function completed` and the TRSAVE `function canceled` responses, the trace service tool function is turned off. The second of these responses is issued only when DASD was specified on the TRSAVE ON command. (It is not issued for CP traces to tape.)

## Messages

- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP006E Invalid device type - {*rdev/vdev/ldev*}
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP046E *type rdev* offline
- HCP053E [XAUTOLOG failed for *userid*:] *userid/value* not in CP directory
- HCP140E *type {rdev/ldev}* attached to *userid*
- HCP143E *type rdev* in use by system
- HCP410E CP ENTERED; PAGING ERROR
- HCP422E The same option was specified twice.
- HCP423E The function requested is incompatible with a previously specified operand.
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP479E Traceid *traceid* currently enabled
- HCP1001E An operand is missing for *option*.
- HCP1013E An invalid operand was supplied for *option - operand*.
- HCP1105E Tape *rdev* not attached; tape assigned elsewhere.
- HCP1107E *command* failed; I/O error on tape *rdev*
- HCP1917E Tape device(s) {*rdev[-rdev]|vdev[-vdev]*} attached MULTIUSER.
- HCP6076E TRSAVE is not active.



- HCP6077E TRSAVE is already active.
- HCP6079E Tape drive *rdev* does not support mode *mode*.
- HCP6085E The TRSAVE command cannot be processed because the function is in a transition state.
- HCP6087E DEFERIO FRAMES specified exceeds the amount of frames eligible for DEFERIO traces.
- HCP6089E DEFERIO FRAMES specified causes total DEFERIO frames to exceed the amount of frames eligible for all DEFERIO traces.
- HCP6091I The amount of frames used for DEFERIO trace exceeds the amount of pages that will be saved on DASD.
- HCP6099E DEFERIO trace is invalid for a TYPE GT BLOCK mode trace.
- HCP6525E External Security Manager is unavailable.

## TRSOURCE

---

### Authorization

Privilege Class: C

### Purpose

Use the TRSOURCE command to support definitions and control of I/O, data, and guest tracing.

The TRSOURCE command is functionally organized as follows:

- TRSOURCE ID defines a trace. TRSOURCE ID associates an event (such as execution of a particular piece of code or an interrupt on a specific device) with a name (*traceid*) and, optionally, with a set of other traces (*traceset*).

The TYPE operand of TRSOURCE ID assigns a trace ID to one of the four trace types. These types are:

1. IO (I/O tracing). The IO trace lets you trace all I/O activity for a specific real device or range of real devices. The times, channel programs, SCSWs and data transferred for each I/O operation to the specified devices are made available for data analysis.
  2. DATA (data tracing). The data trace allows the user to define trace entries dynamically to record execution of most code paths in CP. The LOC parameter specifies what instruction, when executed, triggers recording of information. Data link parameters define what information is collected when the trap starts.
  3. GT (guest tracing). The guest trace allows CMS or GCS to record events in a system trace file.
  4. LAN (LAN tracing). The LAN trace allows a virtual switch or guest LAN to be traced. The trace collects the network traffic flowing over the simulated LAN segment. The traffic flowing over the OSA port (in the case of a virtual switch) might also be recorded.
  5. MONITOR CALL instructions placed in code paths that execute in transactional-execution mode will cause the executing transaction to abort if a monitor event is recognized. This restriction affects the use of TRSOURCE GT and the deployment of class 10 MONITOR CALL instructions.
- TRSOURCE ENABLE activates the tracing of individual trace IDs and of trace IDs within trace sets.
  - TRSOURCE DISABLE deactivates the tracing of individual trace IDs and of trace IDs within trace sets.
  - TRSOURCE DROP disables and drops all CP knowledge of individual trace IDs and trace IDs within trace sets.
  - TRSOURCE DISPLAY displays the status resulting from the TRSOURCE commands you have issued.

See [Using the TRSOURCE Command](#) in *z/VM: Diagnosis Guide* for examples that show how TRSOURCE is used.

### Usage Notes

(General for TRSOURCE)

1. All traces are disabled at system termination.
2. The definitions of traces are not saved across system IPLs. Therefore, the definitions of any commonly defined traces should be saved in EXECs.
3. When a new member of an enabled set is defined, that member is not automatically enabled. It must be enabled separately. Attempts to enable trace IDs that are already enabled do not alter anything and do not result in an error message.
4. The definition of what is being traced for each trace ID cannot be altered while the trace ID is enabled. Any trace ID may be altered while it is disabled.

5. Use of the DROP keyword directed toward a trace set removes any record of all the trace IDs in the set. To remove or *ungroup* a member of a set, either assign the trace ID to another, possibly unused, trace set name or assign the trace ID to a trace set name of NULL.
6. The destination of trace data can be altered with the TRSAVE command while the trace is disabled. See [“TRSAVE” on page 2011](#) for details.
7. Enabling a trace for which TRSAVE has not been issued invokes the TRSAVE destination defaults.
8. If an error message is generated for a trace definition, enter **query trsource id traceid** to verify that the trace definition is correct.
9. Data traced from this command is stored as a system data file (SDF), specifically a trace file (TRF), as opposed to a printer or reader file. Use the TRACERED utility to merge and format the files. TRACERED can send the output to your virtual printer, to a CMS file, or to an OS QSAM file. For more information, see [“TRACERED” on page 2244](#).
10. If an external security manager is installed on your system, you may not be authorized to issue this command. For additional information, contact your security administrator.
11. Instructions whose simulation as part of TRSOURCE would be disruptive are classified as TRSOURCE-ineligible. These include LGG, LLGFSG, PPA, TABORT, TBEGIN, TBEGINC, and TEND.
12. Several instructions are treated as NOPs by TRSOURCE, including BPP, BPRP, and NIAI.

## Examples

The following examples provide an overview of the ways TRSOURCE can be used. For further examples, see [z/VM: Diagnosis Guide](#).

1. This example defines and enables a trace ID of type DATA. The trace ID is DSPLOOP, it is defined at address 4B248, and the data link string specified is G4+8%.2. 'label' is a label that is included with the datalink data in the trace record to identify that datalink's data. The destination is specified as a system trace file for the issuer with a maximum size of 2000.

```
trsource id dsploop type data loc 4b248 5820c846
trsource id dsploop dl g4+8%.2=label
trsave id dsploop dasd to * size 2000
trsource enable id dsploop
```

2. This example defines, enables, and disables two trace IDs of type IO. These trace IDs, IO1 and IO2®, are put into trace set IOSET. IO1 traces I/O to devices in the range C41-C47; up to 200 bytes of transferred data is traced. IO2 traces I/O for USER1 to devices in the range A81-A87; up to 400 bytes of transferred data is traced. The destination options default (because TRSAVE was not issued) and the trace files are named IO1 and IO2 with a maximum size of 256 pages and a maximum number of 2.

```
trsource id io1 set ioset type io
dev c41-c47 iodata 200

trsource id io2 set ioset type io
dev a81-a87 iodata 400 user user1

trsource enable set ioset
trsource disable id io1
trsource disable id io2
```

3. This example defines, enables, and drops two trace IDs of type GT. These trace IDs, GT1 and GT2, are put into trace set GTSET. The destinations are specified as system trace files for user ID USER3 with a default size of 256 pages.

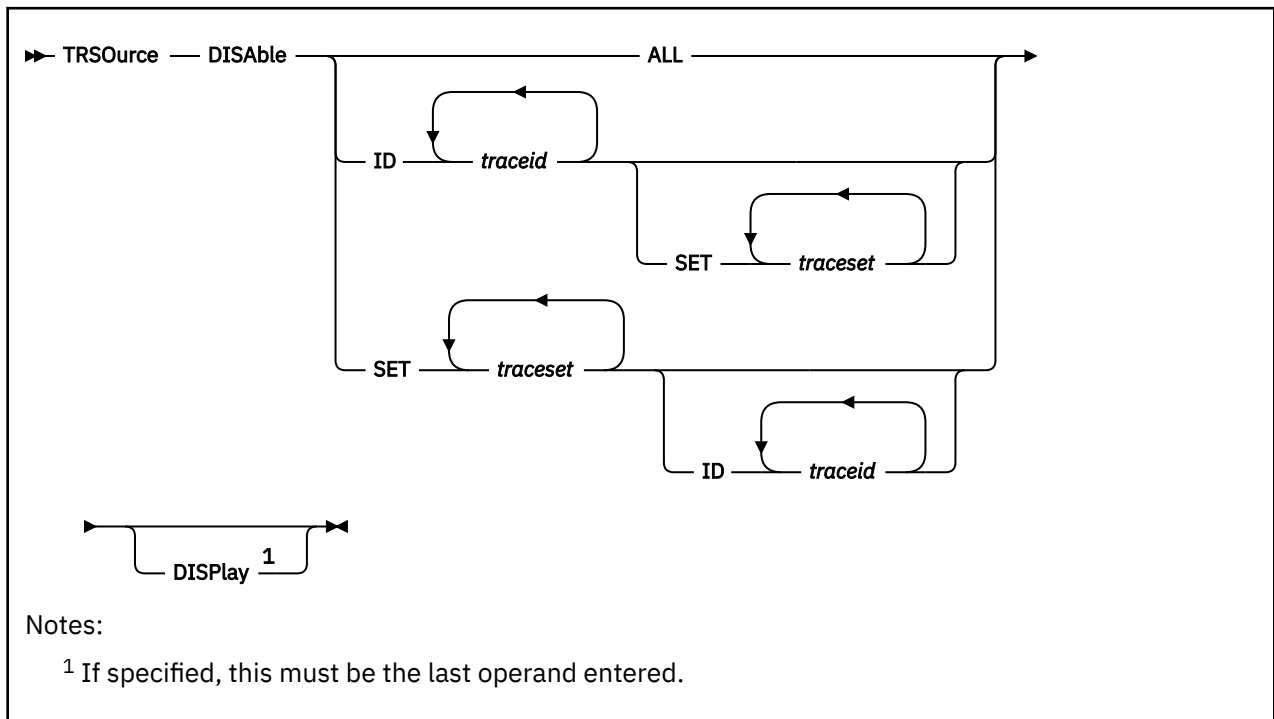
```
trsource id gt1 set gtset type gt for user user2
trsource id gt2 set gtset type gt for vmgroup gcsgroup
trsave id gt1 dasd to user3
trsave id gt2 dasd to user3
trsource enable id gt2
trsource enable id gt1
trsource drop set gtset
```

## Messages

- HCP002E Invalid operand - *operand*
- HCP004E Invalid hexloc - *hexloc*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP045E *userid* not logged on
- HCP289I {User|Group} {*userid|group*} already enabled by traceid *traceid*.
- HCP347E Excessive datalinks supplied for traceid *traceid*
- HCP348E Excessive data requested for traceid *traceid*.
- HCP349E String supplied does not match storage contents.
- HCP369E Invalid location for tracing; command rejected
- HCP370E Invalid datalink string supplied - *value*
- HCP371I Device *rdev* already enabled by traceid *traceid*.
- HCP371I Logical device *ldev* already enabled by traceid *traceid*.
- HCP371I PCIF *rpfid* already enabled by traceid *traceid*
- HCP427E *type* {*rdev|vdev|name[userid]*} system spool space full; file {closed|held|purged}
- HCP428E *type* {*rdev|name[userid]*} spool error; file {held|purged|closed}
- HCP429E *type* *vdev* spool error; file {held|purged}
- HCP439E {User *userid*|\*NSS|\*IMG|\*UCR|\*NLS} spool fileid limit exceeded
- HCP477E Traceid *traceid* not enabled; incomplete definition
- HCP479E Traceid *traceid* currently enabled
- HCP568E Maximum number of devices exceeded for traceid *traceid*.
- HCP644E Unknown/unauthorized CP command: *command*
- HCP816I {Traceid|Traceset} *operand* does not exist
- HCP819I Traceid *traceid* disabled; group *group* enabled by *traceid*
- HCP847E Maximum system spool file limit exceeded
- HCP1003E {*CP module name|rdev|ldev*} {is|was} not found.
- HCP1100E A logical device number was not supplied or it is invalid.
- HCP6081I Ten trace files have been purged for {CP|traceid *traceid[userid]*}. Further warnings are suppressed.
- HCP6084I The maximum number of trace files for {CP|traceid *traceid[userid]*} has been exceeded. The oldest file is purged.
- HCP6086I Traceid *traceid1* is no longer superseded by VMGROUP trace *traceid2*.
- HCP6088I Trace data for traceid *traceid[userid]* was not saved by TRSAVE.
- HCP6089E DEFERIO FRAMES specified causes total DEFERIO frames to exceed the amount of frames eligible for all DEFERIO traces.
- HCP6091I The amount of frames used for DEFERIO trace exceeds the amount of pages that will be saved on DASD.
- HCP6093E You are not allowed to switch modes for traceid *traceid*.
- HCP6094I User *userid* will not be traced by trace *traceid*.
- HCP6095I User *userid* is no longer being traced by trace *traceid*.
- HCP6096I Trace *traceid* is being disabled.

- HCP6097I User *userid* has restricted tracing to traces defined as BLOCK.
- HCP6098E Trace *traceid* can not be enabled.
- HCP6099E DEFERIO trace is invalid for a TYPE GT BLOCK mode trace.
- HCP6160E *token1* was not expected; *xxxxx* was expected.
- HCP6161E Constant definition incorrect: *reason*.
- HCP6162E Maximum nesting depth of IF statement exceeded.
- HCP6163E Internal error during command processing: *module + offset*
- HCP6426I {Allocating|Writing} the DEFERIO frames for traceid *traceid*.
- HCP6427I X% of the DEFERIO frames for traceid *traceid* have been {allocated|written}.
- HCP6525E External Security Manager is unavailable.

## TRSOURCE DISABLE



### Authorization

Privilege Class: C

### Purpose

Use TRSOURCE DISABLE to end tracing of individual trace IDs and tracing of trace IDs within trace sets.

**Note:** The trace file containing the trace data gathered by the trace ID is closed when the trace ID is disabled.

### Operands

#### ID *t1...*

specifies the trace IDs to be deactivated.

#### SET *s1...*

specifies the trace sets to be deactivated.

#### ALL

deactivates all existing trace IDs.

#### DISPlay

displays the status for each trace ID or trace set specified on this command. If you specify DISPLAY, it must be the last operand entered.

### Usage Notes

1. For usage information, see TRSOURCE [“Usage Notes”](#) on page 2018. For disabling a DEFERIO trace, see TRSAVE Usage Note [“17”](#) on page 2014.

## TRSOURCE DISPLAY

---

▶▶ TRSource — DISPlay ▶▶

### Authorization

Privilege Class: C

### Purpose

Use TRSOURCE DISPLAY to display the status of trace IDs and trace sets.

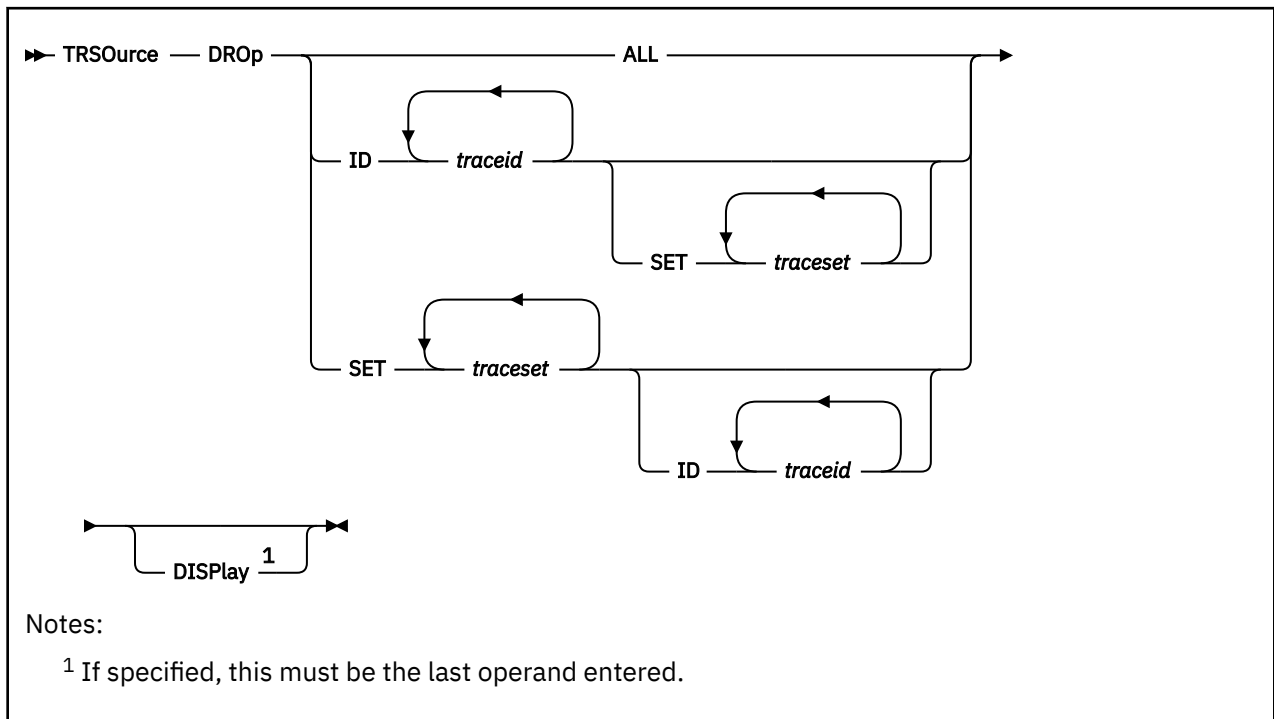
### Usage Notes

1. TRSOURCE DISPLAY provides the same information as QUERY TRSOURCE ALL. For more information about these responses, see [“QUERY TRSOURCE”](#) on page 1239.
2. To display the status of a particular trace ID or trace set, include the DISPLAY operand at the end of a TRSOURCE command. For example, if you enter **trso id me type gt for user operator display**, you receive the following output:

ID	TYPE	SET	STATUS	ORIGINID
ME	GT	NULL	DISABLED	OPERATOR

3. For more information, see TRSOURCE [“Usage Notes”](#) on page 2018.

## TRSOURCE DROP



### Authorization

Privilege Class: C

### Purpose

Use TRSOURCE DROP to disable the specified trace ID or trace sets and delete from CP all individual trace IDs and the trace IDs within trace sets.

### Operands

#### ID *t1...*

specifies the trace IDs to be disabled and deleted.

#### SET *s1...*

specifies the trace sets to be disabled and deleted.

#### ALL

disables and deletes all existing trace IDs regardless of whether they are enabled.

#### DISPlay

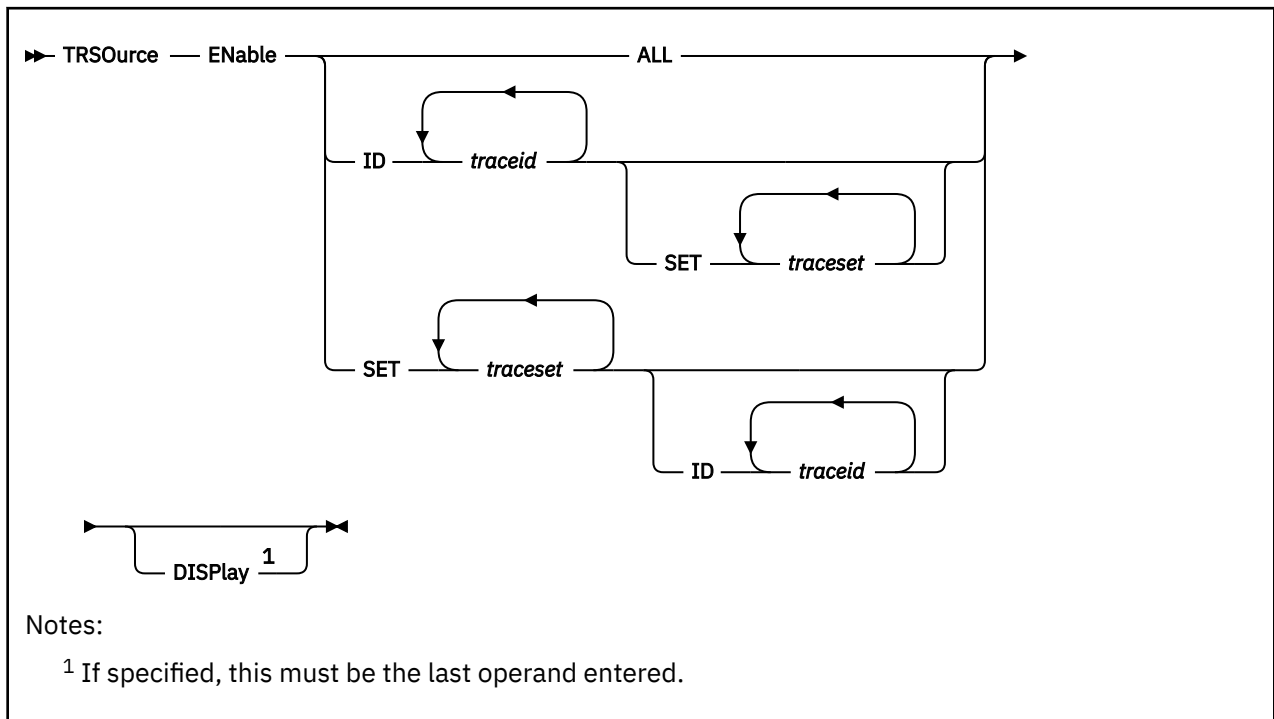
displays the status for each trace ID or trace set specified on this command. If you specify DISPLAY, it must be the last operand entered.

### Usage Notes

1. The trace IDs or trace sets cannot be reenabled without being redefined.
2. Use of the DROP keyword directed toward a trace set removes any record of all the trace IDs in the set. To remove or *ungroup* a member of a set, the user should either assign the trace ID to another, possibly unused, trace set name or assign the trace ID to a trace set name of NULL.
3. For more information, see TRSOURCE “Usage Notes” on page 2018.



## TRSOURCE ENABLE



### Authorization

Privilege Class: C

### Purpose

Use TRSOURCE to activate the trace of individual trace IDs and of trace IDs within trace sets.

### Operands

#### ID *t1...*

specifies the trace IDs to be activated.

#### SET *s1...*

specifies the trace sets to be activated.

#### ALL

activates all existing trace IDs.

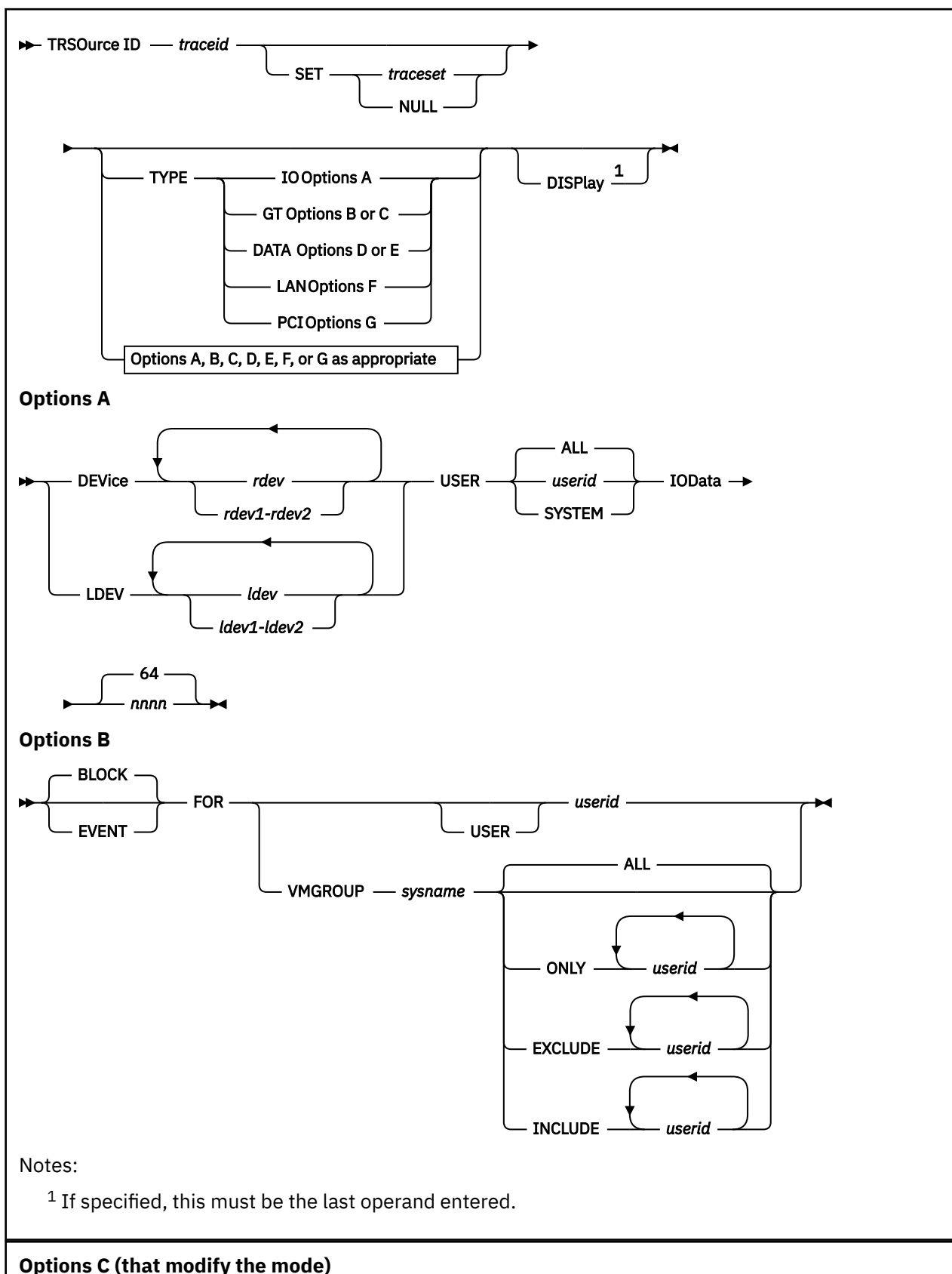
#### DISPlay

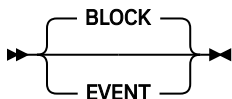
displays the status for each trace ID or trace set specified on this command. If you specify DISPLAY, it must be the last operand entered.

### Usage Notes

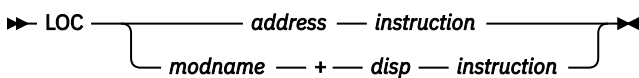
1. For usage information, see TRSOURCE “Usage Notes” on page 2018. For enabling a DEFERIO trace, see TRSAVE usage note “17” on page 2014.

## TRSOURCE ID (Definition Operands)

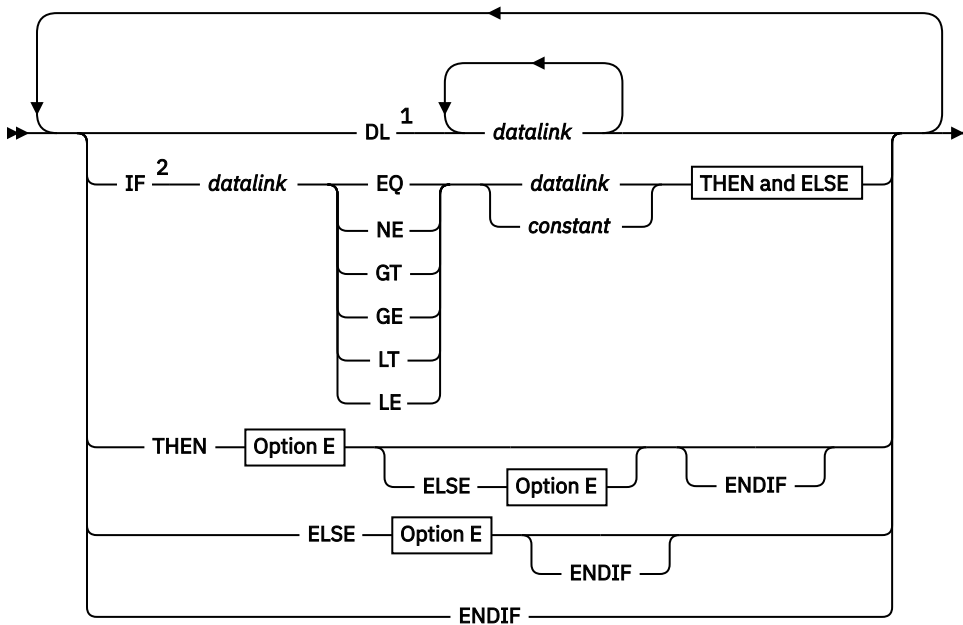




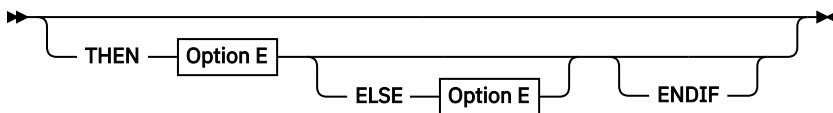
**Options D**



**Options E**



**THEN and ELSE**

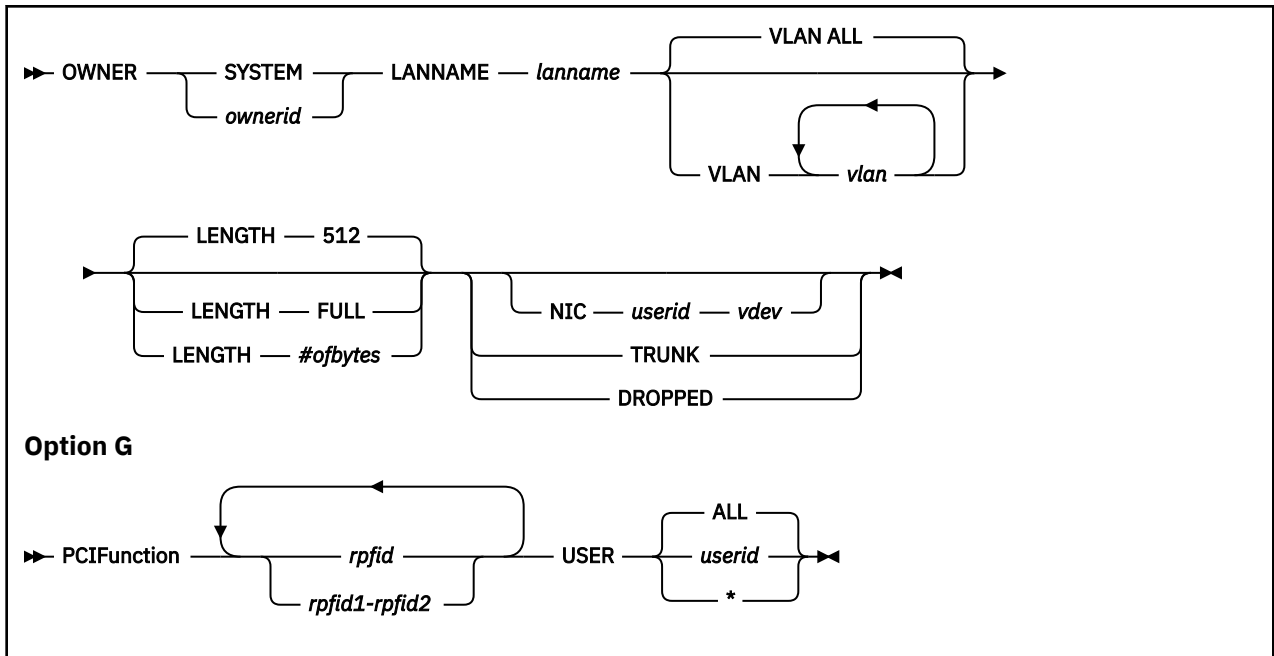


**Notes:**

<sup>1</sup> Use of the DL, IF, THEN, ELSE, and ENDIF keywords is dependent on the keywords that have already been specified. For example, 'THEN' can not be specified unless an 'IF' phrase immediately precedes it. Likewise, 'DL' could not be specified if a 'THEN' is expected. See the "Examples" section below.

<sup>2</sup> IF statements may be nested up to 16 levels deep.

**Option F**



## Authorization

Privilege Class: C

## Purpose

Use TRSOURCE ID to define trace traps—that is, to associate a tracing condition with a name (*traceid*) and, optionally, with a set of other traces (trace set). Also, use the TYPE operand to assign a trace ID to one of the trace types (I/O, data, guest, LAN, or PCI tracing).

You do not need to specify all of the definition operands on each invocation of the TRSOURCE command; you can build the definition piecemeal on a number of TRSOURCE commands. Note, however, that the TYPE operand and certain TYPE parameters must be associated with a trace ID before that trace can be enabled.

## Operands

### *traceid*

identifies the trace; it is a user-defined, alphanumeric, 1- to 8-character string. You can specify *traceid* as a target on the ENABLE, DISABLE, and DROP operands to activate, deactivate, or delete a particular trace; you can also use *traceid* as a target for the TRSAVE command, which can alter the recording options for a trace.

Used with the SET operand, trace IDs can be grouped to form trace sets. A *traceid* cannot be a member of more than one trace set at a time.

The trace ID also serves as the default name of the system trace file for the trace data. The operand trace IDs cannot duplicate any of the following TRSOURCE keywords or any accepted abbreviation of these keywords: ID, SET, TYPE, ENABLE, DISABLE, DROP, DISPLAY, and ALL.

### SET *traceset*

#### SET NULL

is a user-defined, alphanumeric, 1- to 8-character string; it identifies the trace set that includes the specified trace ID. Creating trace sets lets you group individual traces that are functionally or logically related. You can then specify *traceset* as the target of the ENABLE, DISABLE, and DROP operands to activate, deactivate, or delete all of the trace IDs in *traceset*.

A trace set can contain trace IDs of more than one type. A trace set of NULL indicates that the trace ID is not a member of any specific set. Set operations cannot be performed on the NULL set.

A trace ID cannot be made a member of a set while it is enabled. Trace sets cannot duplicate any of the following TRSOURCE keywords or any accepted abbreviation of these keywords: ID, SET, TYPE, ENABLE, DISABLE, DROP, DISPLAY, and ALL.

**TYPE IO****TYPE GT****TYPE DATA****TYPE LAN****TYPE PCI**

specifies a system-defined trace type. The trace type determines which parameters are allowed and which are required for trace ID. The TYPE operand need only be specified once for a given trace ID but must be specified before or at the same time as any type parameter for that trace ID.

Different trace IDs of the same trace type can be placed in the same or different trace sets. Trace types cannot be used as targets of the ENABLE, DISABLE, and DROP operands. Grouping of trace IDs is done with the SET operand, not the TYPE operand.

***type parameters***

The parameters and descriptions for the respective trace types are documented in the following sections:

- "The I/O Trace"
- "The GT Trace"
- "The Data Trace"
- "The LAN Trace"
- "The PCI Trace"

**DISP**

displays the status for each trace ID or trace set specified on this command. If you specify DISPLAY, it must be the last operand on the command line.

**The I/O Trace**

The I/O trace lets you trace all I/O activity on a user-specified real or logical device, a list of such devices, or range of such devices. The times, channel programs, SCSWs, and data transferred for each I/O operation to the specified devices are made available for data analysis. If concurrent sense data is received then this information is made available as well.

**Note:** One of the following I/O parameters must be specified on a TRSOURCE command.

```
DEVICE  rdev1[-rdev2] [rdev3[-rdev4]]...
LDEV    ldev1[-ldev2] [ldev3[-ldev4]]...
```

**DEVICE rdev1...**

specifies the device number or ranges of device numbers whose I/O activity is to be traced. The number of devices or ranges that can be traced with one trace ID of type IO is limited to 255. A device is a member of only one enabled trace ID. If you re-specify a DEVICE for a previously defined trace, the new device will be appended to the definition. If you attempt to enable the same device from two or more different trace IDs, you will receive an error message.

**LDEV ldev1...**

specifies the logical device number or ranges of logical device numbers whose I/O activity is to be traced. A logical device number is comprised of an optional letter "L" followed by one to four hexadecimal digits. The number of logical devices or ranges that can be traced with one trace ID of type IO is limited to 255. A logical device is a member of only one enabled trace ID. If you re-specify an LDEV for a previously defined trace, the new logical device will be appended to the definition. If you attempt to enable the same logical device from two or more different trace IDs, you will receive an error message.

**Note:** Any or none of the following I/O parameters may be specified in any order on a TRSOURCE command.

```

USER      { userid }
          { ALL   }
          { SYSTEM }

IOData    { nnnn }
          { 0    }
    
```

**USER *userid***

**USER ALL**

**USER SYSTEM**

specifies which user's I/O should be traced. I/O done by the system on behalf of the user is traced. Tracing can only be done for one user per trace ID and any respecification of the user parameter replaces the existing definition. The user ID SYSTEM is a valid user ID for I/O traces. Only one user's I/O or all I/O is traced. The default value, ALL, specifies that all I/O to the specified device is traced for all user IDs.

**IOData *nnnn***

specifies how many bytes, if possible, of the transferred data should be traced. The valid range for *nnnn* is 0 to 3800 decimal and the default is 64 bytes.

The GT Trace

Traces of type GT allow a guest virtual machine to record events. Guests can record in block or event mode. All user IDs specified for a trace ID must support the same mode of recording.

```

[BLOCK] FOR { [USER] userid
[EVENT]      {
                VMGROUP sysname [ALL
                                [ONLY u1 u2... ]
                                [EXCLUDE u1 u2... ]
                                [INCLUDE u1 u2... ]
                }
    
```

**BLOCK**

**EVENT**

specifies the recording mode for tracing. This mode must match the capabilities of the guest virtual machine.

If you specify BLOCK, the guest virtual machine must be able to intercept the monitor call program interrupt, collect the trace entry data in a buffer, and then pass blocks of trace data to the control program. These blocks are written to a system trace file (TRFILE). BLOCK is the default.

If you specify EVENT, the virtual machine uses the monitor call class 10 interface (CP intercepts the MC program interrupt) to pass individual records directly to the control program. Each trace record is written to a system trace file.

EVENT trace records will not be created when the guest uses MC class 10 with the enhanced-monitor facility and a monitor-event-counting operation is performed.

See *z/VM: Diagnosis Guide* for examples.

**FOR USER *userid***

specifies the user for which trace events are recorded.

**FOR VMGROUP *sysname***

specifies the VMGROUP for which trace events are recorded.

**ALL**

specifies that every virtual machine in the VMGROUP *sysname* is included in the defined GT trace. This is the default.

**ONLY**

specifies that the virtual machines included in the defined GT trace are limited to user IDs *u1 u2.....*

**EXCLUDE**

specifies that every virtual machine in the VMGROUP *sysname* is included in the defined GT trace except those virtual machines with *userid*s *u1 u2...*

**INCLUDE**

specifies that user IDs *u1 u2...* should be added to those virtual machines that currently comprise the VMGROUP *sysname* in the defined GT trace.

After a VMGROUP trace has been defined, parameters may be specified to allow further changes in the group of virtual machines that comprise the GT trace. The VMGROUP trace is identified with the ID parameter. These modifications can be entered at any time; the trace specified does not need to be disabled.

Parameters may be specified to change the mode of the TYPE GT trace before it is enabled. The mode must match the capabilities of the guest virtual machine.

*Type GT Parameters that Modify the Mode:*

```
[BLOCK]
[EVENT]
```

**BLOCK**

**EVENT**

specifies the recording mode for tracing. This mode must match the capabilities of the guest virtual machine.

If you specify BLOCK, the guest virtual machine passes blocks of trace data to the control program. These blocks are written to a system trace file.

If you specify EVENT, the virtual machine passes individual records directly to the control program. Each trace record is written to a system trace file.

The DATA Trace

Trace type DATA allows the user to dynamically define trace entries to record execution of most code paths in CP. The LOC parameter specifies which instruction, when executed, triggers recording of information. Data link parameters define what information is collected when the trace springs.

**Note:** Any or none of the following DATA parameters may be specified on a TRSOURCE command. Specification of a given parameter must be consistent with the state of the traceid as defined by previously entered parameters, including parameters entered via previous invocations of the TRSOURCE command.

```
LOC  { address instruction }
     { modname + disp instruction }

DL   datalinks

IF   datalink { EQ } { datalink }
     {          } { constant }
     {          } {          }
     {          } {          }
     {          } {          }
     {          } {          }
     {          } {          }

THEN { DL datalinks }
     {          }
     IF ... ENDIF

ELSE { DL datalinks }
     {          }
     IF ... ENDIF

ENDIF
```

**LOC address**

specifies the address in the CP nucleus that identifies the trace point at which the data is to be collected.

**LOC modname + disp**

specifies the CP module name and displacement into that module that identifies the trace point at which the data is to be collected. If the module is written in a higher-level language, an 8-character module name must be supplied.

See Type Data Usage Note [“6” on page 2037](#) for a list of restricted locations.

**instruction**

identifies in hexadecimal characters the contents of storage at the specified trace point. Matching the specified instruction with the contents of storage verifies that the trace point is at the intended location in the CP nucleus. The instruction supplied must be at least as long as the instruction to be verified. Only that instruction is verified, and any extraneous hexadecimal data is ignored. If the verification fails, the LOC parameters are not accepted, and a message is sent to the user who entered the command.

**DL datalinks**

identifies the data to be traced. The following character groups can be used to make up a data link string:

1. Addresses: substrings consisting of characters 0 to 9 and A to F.
2. Registers: substrings beginning with G or GG (for general purpose registers) or X or XG (for control registers) followed by a hexadecimal digit. A range of registers may be specified by using a colon between the two registers, such as *GO:4*.
3. Indirection: a percentage sign (%) or an ampersand (&). Indirection means that the contents of an address are used instead of the address itself. A percentage sign uses a four-byte indirect address; an ampersand uses an eight-byte indirect address.
4. Address type: L or R. L indicates that the address derived from the preceding expression, and all subsequently derived addresses, should be treated as host logical storage addresses. R indicates that the address derived from the preceding expression, and all subsequently derived addresses, should be treated as host real storage addresses. For each Datalink, addresses are treated as host logical storage addresses until the first occurrence of R, if any.
5. Arithmetic: a plus sign (+) or a minus sign (-).
6. Length delimiter: a period (.).
7. Mnemonic marker: an equals sign (=).

The data link strings specify how to combine these groups to reach the data to be traced. The following is a Backus-Naur definition of the syntax of data link strings:

```

<datalink> ::= <main> | <main>=<mnemonic>
    <main> ::= <anchor> | <anchor><vector> | <anchor>.<length> |
              <anchor><vector>.<length> | <range>
    <anchor> ::= <reg> | 0..FFFFFFFF
    <vector> ::= <modifier> | <vector><modifier>
    <modifier> ::= +<d> | -<d> | +<reg> | -<reg> | % | & | R | L
    <length> ::= 0..F00
    <range> ::= G<digit>:<digit> | X<digit>:<digit> | GG<digit>:<digit>
              | XG<digit>:<digit>
    <reg> ::= G<digit> | X<digit> | GG<digit> | XG<digit>
    <digit> ::= 0..15 | 0..9, A, B, C, D, E, F
    <d> ::= 0..7FFF
    <mnemonic> ::= mmmmmmm
    
```



where mmmmmmm indicates a user-generated 1- to 8-character alphanumeric string.

Data can be identified by either using the current register contents as pointers or knowing the real location of the data to be traced. Data links define the procedure to be used to locate the desired data.

**Note:** To interpret Backus-Naur Form, the symbol colon colon equal (::<=) may be interpreted as “can be specified as”. Thus, a statement such as

```
<location> ::= <register> | <address>
```

can be read as “Location can be specified as register or address”. Any symbol on the right-hand side of the colon colon equal (::<=) can be defined on another statement in the Backus-Naur Form.

#### **IF datalink**

identifies the data to be used as the first operand for data comparison that determines whether a subsequent THEN statement or ELSE statement is executed.

Up to 16 nested IF statements are allowed. Each IF statement must be paired with an ENDIF statement.

#### **EQ datalink**

#### **NE datalink**

#### **GT datalink**

#### **GE datalink**

#### **LT datalink**

#### **LE datalink**

identifies the compare operation to be performed and identifies the data to be used as the second operand for data comparison that determines whether a subsequent THEN statement or ELSE statement is executed.

The allowable conditional operators are:

#### **EQ**

(first operand equal to second operand),

#### **NE**

(first operand not equal to second operand),

#### **GT**

(first operand greater than second operand),

#### **GE**

(first operand greater than or equal to second operand),

#### **LT**

(first operand less than second operand),

#### **LE**

(first operand less than or equal to second operand).

The comparison is a logical compare. The length of the data compared is the lesser of the length determined by the first operand datalink or by the second operand datalink.

#### **EQ constant**

#### **NE constant**

#### **GT constant**

#### **GE constant**

#### **LT constant**

#### **LE constant**

identifies the compare operation to be performed and identifies a constant to be used as the second operand for data comparison that determines whether a subsequent THEN statement or ELSE statement is executed.

The allowable constants are:

**X'...';**

where ... are an even number of up to 16 hexadecimal digits (0 to 9 and A to F), or

**C'....';**

where .... are up to 64 characters. Blanks and single quotation marks may be used in the constant string. Use two consecutive single quotation marks to specify one single quotation.

The comparison is a logical compare. The length of the data compared is the lesser of the length determined by the first operand datalink or by the (second operand) constant.

**THEN DL *datalinks***

identifies the data to be traced if the result of the preceding IF statement is true. A THEN statement may be followed by any number of datalinks, all of which are executed only if the result of the preceding IF statement is true.

A subsequent IF statement (not preceded by an ELSE statement or ENDIF statement; that is, a nested IF statement) would be executed only if the result of the preceding IF statement is true.

**THEN IF**

identifies a nested IF statement. Such an IF statement would be executed only if the result of the preceding IF statement is true.

Up to 16 nested IF statements are allowed. Each IF statement must be paired with an ENDIF statement.

**ELSE DL *datalinks***

identifies the data to be traced if the result of the preceding IF statement is false. An ELSE statement may be followed by any number of datalinks all of which are executed only if the result of the preceding IF statement is false.

A subsequent IF statement (not preceded by an ENDIF statement, thus a nested IF statement) would be executed only if the result of the preceding IF statement is false.

**ELSE IF**

identifies a nested IF statement. Such an IF statement would be executed only if the result of the preceding IF statement is false.

Up to 16 nested IF statements are allowed. Each IF statement must be paired with an ENDIF statement.

**ENDIF**

identifies the end of conditional datalinks associated with the last unclosed IF statement.

Each IF statement must be closed with an ENDIF statement before the traceid can be enabled.

The LAN Trace

The LAN trace type allows a virtual switch or guest LAN to be traced. The trace collects the network traffic flowing over the simulated LAN segment. When a trace is defined with the default options, all traffic on the specified LAN will be traced regardless of VLAN ID. A total of 512 bytes per datagram will be collected and discarded as well as delivered datagrams will be recorded. The traffic flowing over the OSA port (in the case of a virtual switch) will also be recorded. The following options are valid on a LAN trace:

**OWNER *ownerid***

specifies the owner of the guest LAN or virtual switch to be traced. The owner of a virtual switch is always SYSTEM. The owner id must be a valid user ID or SYSTEM.

**LANNNAME *lanname***

specifies the name of the guest LAN or virtual switch to be traced. The LAN must be defined before the trace id is enabled.

**VLAN *vlanid***

**VLAN ALL**

specifies which vlan traffic should be traced. Up to four different vlan ids can be specified. In order to trace all traffic regardless of vlan id, specify VLAN ALL.

**LENGTH *length*****LENGTH 512****LENGTH FULL**

specifies how many bytes of the packet should be traced. The default is 512 bytes. If FULL is specified, a total of 2048 bytes will be recorded. The specified LENGTH can be between 64 and 2048.

**NIC *userid vdev***

specifies that only traffic inbound or outbound for the specific *userid vdev* pair will be traced. This option allows the data collection to be filtered for a specific guest connection. The *vdev* specified should be the address of the data device.

**TRUNK**

specifies that only traffic flowing over the virtual switch OSA (RDEV) connection will be traced. If TRUNK is specified on a guest LAN, no data is collected.

**DROPPED**

specifies that only dropped packets will be collected. A dropped packet is a datagram that could not be delivered to a host or datagram that had an error, such as a missing header causing the packet to be too short. Without this option, dropped packets and successful packets are both recorded.

For more information on [CP TRSOURCE TYPE LAN Command](#), see *z/VM: Connectivity*.

The PCI Trace

The PCI trace type allows I/O operations on a PCI function to be traced. The following options are valid on a PCI trace:

**Note:** The following PCI parameter must be specified on a TRSOURCE command.

```
PCIFunction rpfid1[-rpfid2] [rpfid3[-rpfid4]]...
```

**PCIFunction *rpfid1*...**

specifies a real PCI function ID or a range of PCI function IDs whose I/O activity is to be traced. Specify a PCI function ID as 8 hexadecimal digits. You may omit leading zeros. The number of real PCI functions or ranges of real PCI functions that can be traced with one trace ID of type PCI is limited to 255. A function is a member of only one enabled trace ID. If you re-specify a PCI for a previously defined trace, the new *rpfid* will be appended to the definition. If you attempt to enable the same *rpfid* from two or more different trace IDs, you will receive an error message.

**Note:** Any or none of the following PCI parameters may be specified in any order on a TRSOURCE command.

```
USER      { userid }
           { ALL   }
           { SYSTEM }
```

**USER *userid*****USER ALL****USER SYSTEM**

specifies which user's I/O should be traced. I/O done by the system on behalf of the user is traced. Tracing can only be done for one user per trace ID and any re-specification of the user parameter replaces the existing definition. The user ID SYSTEM is a valid user ID for I/O traces. Only one user's I/O or all I/O is traced. The default value, ALL, specifies that all I/O to the specified function is traced for all user IDs.

**Usage Notes**

(General for TRSOURCE ID)

1. If you use the ID operand to define a trace ID, ID must be the first operand on the command line.

2. While specific type-dependent operands are optional on the TRSOURCE command, certain type-dependent operands may be required for a trace ID before it can be enabled. Refer to the description of the specific trace types to determine which operands are required before a trace ID of that trace type can be enabled.
3. The TYPE operand, if used, must directly precede any type-dependent operands.
4. All type-dependent operands must be contiguous. The order depends on the requirements of the trace type.

### Type I/O Usage Notes

1. You can define an I/O trace for a nonexistent device—that is, a device that has no RDEVICE definition. If a trace with *only* such devices is enabled, message HCP477E is issued:

```
Traceid (TRACEid) not enabled; incomplete definition
```

Note, however, that a trace with a nonexistent device and a real device (whether the latter be online or offline) *can* be enabled, and the response to QUERY TRSOURCE indicates that a trace for the nonexistent device is activated.

2. System trace files reside on system spool space. Tracing I/O to system spool space causes trace recursion and may cause severe performance degradation, especially for large values of IODATA.

### Type GT Usage Notes

1. Specification of a TYPE GT trace for a VMGROUP makes it easier to control and define guest traces for the members of a group.

A trace TYPE GT for a VMGROUP is used to sift TRSOURCE definition and TRSAVE destination information from the VMGROUP to the virtual machines that are members of the VMGROUP. (The system name specified on the VMGROUP parameter is the name of the named saved system that was IPLed by its members.)

2. Multiple TYPE GT traces can be defined for each virtual machine. Only one TYPE GT trace can be enabled for each virtual machine.
3. A TYPE GT USER trace for a virtual machine that has been defined before the virtual machine becomes a member of a VMGROUP for which a TYPE GT trace is enabled is automatically disabled when the virtual machine becomes a member of the VMGROUP. The USER TYPE GT trace remains in its disabled state after the virtual machine either leaves the VMGROUP, is removed from the VMGROUP by a later TRSOURCE command that excludes that virtual machine's user ID from the VMGROUP trace or the VMGROUP trace is disabled. A TRSOURCE ENABLE command is required to reenble the USER TYPE GT trace.
4. If you need to redefine the destination of the trace data for one of the members of a VMGROUP trace, you must exclude the virtual machine's user ID from the VMGROUP trace and then enable a TYPE GT USER trace containing the desired destination information for the virtual machine.
5. A message is sent to the user who defined a USER TYPE GT trace whenever a defined and enabled USER TYPE GT trace is disabled because of VMGROUP TYPE GT trace activity for a VMGROUP member.
6. A message is sent to the user who defined a USER TYPE GT trace whenever a virtual machine that is a member of a VMGROUP trace is excluded from the VMGROUP by a later TRSOURCE command. This user is also notified if the virtual machine was a member of a VMGROUP for which tracing was disabled.
7. Use the BLOCK or EVENT keyword to tell CP whether the guest virtual machine will collect all the guest trace data in a buffer. Unpredictable results occur when TRSOURCE ENABLE is issued if the trace setup does not match the capabilities of the guest virtual machine.
8. If specified, the BLOCK or EVENT keyword must precede other TYPE GT parameters.
9. The mode, BLOCK or EVENT, may not be changed after a trace is enabled; even if it has been disabled. You must cancel the trace first.

10. A virtual machine may declare to CP that it will be sending trace data in BLOCK mode. CP rejects any TRSOURCE ENABLE command that specifies EVENT mode.

#### Type DATA Usage Notes

1. Data collection for trace IDs of trace type DATA occurs before execution of the overlaid instruction. If no data links are specified for the trace ID, a trace entry is created in the system trace file, but no data is collected.
2. To calculate the exact amount of data that can be collected in a trace record, use the following formula:  

$$(3 \text{ times the number of data links specified}) + (\text{the total lengths of the data links}) + (\text{the total number of bytes to be collected}).$$

The sum must be less than or equal to 4 KB (in decimal).

Also, the maximum number of data links allowed for any given trace ID is 255. The maximum data link length is 250.
3. DATA traces should always be placed at instruction boundaries. The TRSOURCE command can verify only that the specified location is on a halfword boundary and that it contains a valid operation code; if these conditions are not met, the command fails, with an error message.  

If the verification is successful, the TRSOURCE command overlays the specified location with a data trace SVC instruction. If the location is not an instruction boundary, unpredictable results can occur, including a system failure.
4. If an error is made while entering a data link, the entire trace must be dropped and redefined to remove the data link from the definition. This is because there is no way of deleting a data link from a trace definition. It is therefore recommended that the definitions of commonly used type DATA traces be saved in EXECs. Locations can be changed for defined but disabled DATA traces.
5. Specification of extra data links for a defined but disabled trace results in an addition to the definition.
6. The following is a list of restricted locations for the definition of trace points. These locations are restricted because enabling DATA traces in them would cause recursion or interlocks.
  - a. Locations not on halfword boundaries.
  - b. Locations not mapped in the system execution space.
  - c. Locations of already enabled DATA traces. Note that multiple DATA traces may be defined at the same location but only one trace at a time may be enabled at that location.
  - d. Locations within a module loaded with the CPXLOAD command.
  - e. Locations that contain a NOOP (4700xxxx) preceded by a Subtract Halfword instruction.
  - f. Locations that contain an enabled dynamic CP exit.
7. Trace data is not collected during system termination.
8. It is recommended that records defined for any code path traveled when writing a system trace file (including paging and spooling code paths and some code paths in IOS) be kept under a page in length.
9. Users should avoid placing traces in code paths that hold spin locks, as this situation increases the path length under the spin lock and seriously decreases system performance.
10. Users should avoid placing data traces on instructions that are targets of EXECUTE instructions; otherwise, results are unpredictable.
11. If interpretation of a data link string during execution results in a negative or invalid address, a record is added to the system trace file to indicate that the error occurred. However, there is no external indication that the error occurred.
12. To trace data located in the absolute page zero, reverse prefixing must be used. When the value of the prefix register is added to the datalink string, values are obtained from absolute page zero. On the other hand, if the address calculated from a data link is less than hexadecimal 2000, the value recorded is obtained from the prefix page of the processor in which the data trace event occurs.

13. If a traceid contains conditional statements (at least one IF statement) and the condition when the trace point is encountered results in the execution of no datalinks, then no trace record is made, not even a trace record containing only header information. It is as though the trace event never occurred.
14. It is possible to specify lower case characters in a character constant definitions. This cannot be done by typing a CP command because all characters in the command are converted to upper case before the TRSOURCE command parser receives the command. To specify lower case, use the Diagnose X'08' interface from an EXEC or from an assembler program. Only the character constant should be specified in lower case. The following is an example of an EXEC that defines a character constant that contains lower case characters and single quotation marks.

```

/* */
cpcmd="TRSOURCE ID TT TYPE DATA IF G3.20 EQ C'Specify ''vdev'' "
response = ''
parse value Diag('8',cpcmd) with response '15'x
if response =v then say 'response=response'
    
```

**Examples**

**Examples of Data Link Usage:** The following examples are based on the status of the registers and the storage pictured below.

Registers	Host Logical Storage				
G1  00001000	4000	00005008	00000000	0000A000	00000000
	5000	0000A004	00001144	00005555	00000000
	6000	00000000	00000000	0000A000	00000000
G4  00004000	...				
	A000	0000B000	0000C000	0000D000	0000E000
	B000	C4C1E3C3	E3D9C1D7	11110000	00000000
G5  00005000	C000	C4C1E3C3	E3D9C1D7	FFFF0000	00000000
	D000	DDDDDDDD	00000000	00000000	00000000
G6  00006000					

Ex.	Datalink	Data Collected
1)	4000	00005008
2)	4000%	00005555
3)	5000%.8	0000C000 0000D000
4)	G5	00005000
5)	G5.4	0000A004
6)	G5+4	00001144
7)	G5+4%	contents of 1144
8)	G5+G6+8%	contents of 11110000
9)	G5%.C	C4C1E3C3 E3D9C1D7 FFFF0000
10)	G4:6	00004000 00005000 00006000
11)	G6+A.2	A000
12)	G4+8%+4%.A	C4C1E3C3 E3D9C1D7 FFFF
13)	G5%+4	0000D000
14)	G5%+4%	DDDDDDDD
15)	IF G5%.8 EQ G4+8%+4% THEN DL G4+8 ELSE DL G4 ENDIF	0000A000

**Note:** Registers, when specified alone as in example 4 above, do not default to an indirection and a length of 4, as other data link values do. In this case, the data collected is simply the contents of the register.

**Examples of Data Link Definition:** The following example shows how multiple TRSOURCE command invocations are used to set up a conditional data link trace.

```

1) TRSOURCE ID TRC1 TYPE DATA LOC HCPNOS + 34 5840C08C
2) TRSOURCE ID TRC1 DL GB
3) TRSOURCE ID TRC1 IF GB+80.8 EQ C'OPERATOR'
4) TRSOURCE ID TRC1 THEN DL G0:F
5) TRSOURCE ID TRC1 ELSE IF GB+80.8 EQ C'MAINT'
6) TRSOURCE ID TRC1      THEN DL GC.30
7) TRSOURCE ID TRC1      ENDIF
8) TRSOURCE ID TRC1      DL G8
9) TRSOURCE ID TRC1 ENDIF

```

-----

These TRSOURCE commands could be defined in an EXEC file. In this example, line 1 defines where the trace is defined. Line 2 shows that when this instruction in HCPNOS executed, an unconditional data link will show the contents of register 11. Lines 3 through 9 define a conditional data link. Notice that in lines 6 and 8, spaces are used to improve readability.

**Examples of TRSOURCE TYPE LAN:** The following example shows how to set up a trace on a virtual switch and collect only data for vlan id 4, 5, and 7 as well as collecting 1000 bytes of data:

```

CP TRSOURCE ID ex1 TYPE LAN OWNER SYSTEM LANNAME VSWITCH1 VLAN 4 5 7 LENGTH
1000

```

If instead an entire guest LAN had to be traced, the following would be specified:

```

CP TRSOURCE ID ex2 TYPE LAN OWNER BOLINDA LANNAME GLAN

```

The trace can also be more specific and capture only packets related to Guest TCPIP1.

```

CP TRSOURCE ID ex3 TYPE LAN OWNER SYSTEM LANNAME VSWITCH1 NIC TCPIP1 2D02

```

## UNCOUPLE

```
▶▶ UNCOUPLE — vdev ◀◀
```

### Authorization

Privilege Class: G

### Purpose

Use UNCOUPLE to disconnect a virtual CTCA from a coupled CTCA device, or to disconnect a virtual network adapter from a guest LAN or a virtual switch. You can also uncouple a virtual CTCA by using an improper form of the COUPLE command. For example,

```
COUPLE vdev TO * vdev
```

would disconnect *vdev* from the current partner and leave it disconnected because it is not possible to connect a CTCA to itself.

### Operands

#### *vdev*

specifies the virtual device number (hex) of the adapter to be disconnected. You may specify the virtual device number of any device that is defined as part of the adapter. However, CP responses refer to each virtual adapter by its *base* device address; this is the first device in the range.

### Usage Notes

When *vdev* is a virtual OSA device type, it represents a virtual network adapter created by the DEFINE NIC command. You can use any *vdev* on the command line that is part of the virtual adapter configuration. However, CP responses use the base device number to identify the virtual adapter.

### Responses

Response 1: (virtual CTCA uncoupled by partner)

```
CTCA vdev DROPPED BY invoker vdev
```

is the response from a partner's console after a virtual CTCA device is uncoupled by the other user.

Response 2: (virtual CTCA uncoupled by user)

```
CTCA vdev DROPPED FROM userid vdev  
CTCA vdev is uncoupled now
```

is the response from an invoker's console after a virtual CTCA device is uncoupled by the user.

Response 3: (virtual NIC uncoupled)

```
NIC vdev is disconnected from LAN ownerid lanname
```

is the response from the invoker's console after a virtual network adapter is uncoupled.

Response 4: (virtual NIC uncoupled)

```
NIC vdev is disconnected from switch SYSTEM switchnm
```

is the response from the invoker's console after a virtual network adapter is uncoupled from a virtual switch.



**Messages**

- HCP006E Invalid device type - *vdev*
- HCP040E Device *vdev* does not exist

## UNDIAL

---

» UNDIAL «

### Authorization

Privilege Class: Any

### Purpose

Use UNDIAL to logically disconnect your terminal from a z/VM guest system and return control of that terminal to the host z/VM system.

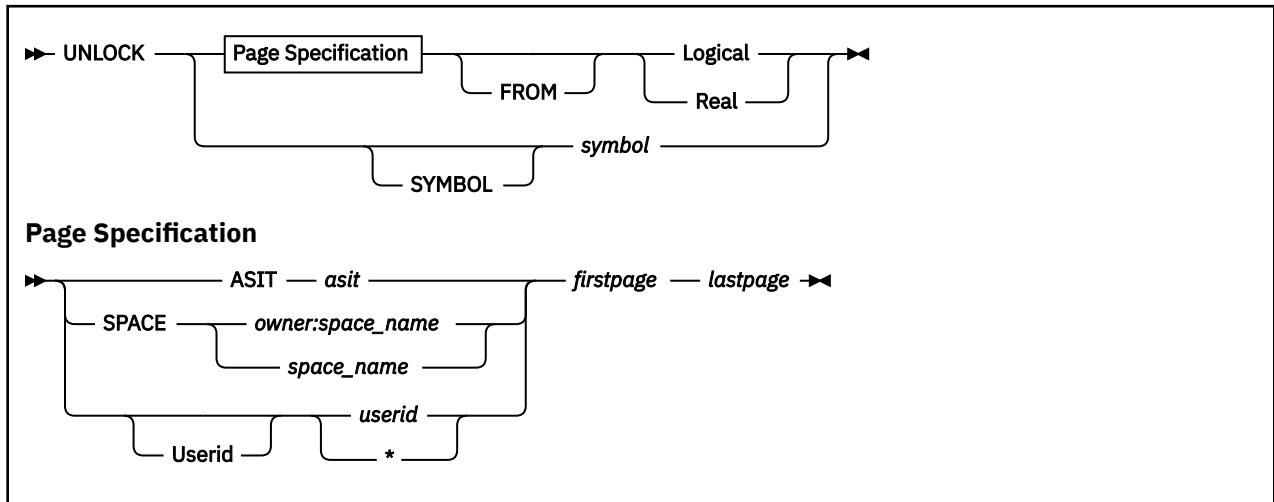
### Usage Notes

1. If UNDIAL is issued from a terminal that is being controlled by a host z/VM system, an error message will be issued.
2. UNDIAL can be issued only from a terminal that is not logged on. That is, UNDIAL can be issued before the LOGON command or after the LOGOFF or DISCONNECT commands.
3. UNDIAL cannot be used if the terminal is connected to a VTAM service machine (VSM) running in the guest system. If this is the case, the connection to the VSM must be terminated before the CP UNDIAL command can be used.
4. When UNDIAL is issued from a terminal that is dialed to the guest VM system as a logical terminal or a non-SNA TTY terminal and is not logged on, UNDIAL has the same effect as the LOGOFF command.

### Messages

- HCP2101E This terminal is not dialed to a virtual machine.
- HCP2102E This terminal is logged on to a virtual machine. You must logoff or disconnect before issuing UNDIAL.
- HCP2103E This terminal is connected to a VTAM Service Machine. You must disconnect from the service machine before issuing UNDIAL.

## UNLOCK



## Authorization

Privilege Class: A

## Purpose

Use UNLOCK to release guest pages kept in host storage by the LOCK command.

## Operands

### ASIT *asit*

unlocks pages in the address space identified by the address-space-identification token, *asit*. The *asit* must be padded to 16 hexadecimal digits with leading zeros.

### SPACE *owner:space\_name*

unlocks pages in the address space identified by the space ID, *owner:space\_name*, where *owner* is the user ID of the virtual machine that owns the address space and *space\_name* is the name assigned to the address space.

If *owner:* is not specified, your virtual machine is assumed to be the owner of the address space.

The space name is a string of alphanumeric and national characters, 1 to 24 characters long. The space name "BASE" is preassigned by CP to be the name of the issuer's host-primary address space.

### USERID *userid*

#### *userid*

unlocks pages in the primary address space of the virtual machine identified by *userid*. The virtual machine must be logged on. If you are unlocking pages of your own virtual machine's storage, you can specify *userid* as an asterisk (\*).

### *firstpage*

is the hexadecimal number of the first user or address space page to be unlocked from host storage.

Usage note [“2” on page 2044](#) explains how to derive page numbers from storage addresses.

### *lastpage*

is the hexadecimal number of the last user or address space page to be unlocked from host storage. If only one page is to be unlocked, *lastpage* must be the same as *firstpage*.

### Logical

unlocks the guest pages from host logical storage pages.

## UNLOCK

### Real

unlocks the guest pages from real storage frames.

### SYMBOL *symbol*

has no effect and is retained only for compatibility purposes. All symbols are considered resident, which means they cannot be locked or unlocked.

## Usage Notes

1. In the primary address space, the first and last page numbers are treated as *guest real* addresses. They are prefixed with the prefix register value of the base virtual CPU of the target virtual machine. In other address spaces, the first and last page numbers are treated as *guest absolute* addresses. Prefixing is not applied to these addresses.
2. Specify page numbers for the *firstpage* and *lastpage* operands, not addresses. Page numbers are assigned beginning with zero; each page is X'1000'. For example, to unlock storage locations X'12000' to X'2CFFF' of USERA's primary address space from host logical storage, enter the following command:

```
unlock usera 12 2c from logical
```

3. The LOCK and UNLOCK commands operate independently of the DIAGNOSE code X'98' page locking facility. Pages locked by the LOCK command cannot be unlocked by DIAGNOSE code X'98', nor can real I/O pages locked by DIAGNOSE code X'98' be unlocked by the UNLOCK command.
4. If you issue the LOCK command to lock a page in a shared segment and do not issue an UNLOCK command, the page is unlocked when the last user of the shared segment releases the storage with the LOGOFF command or another command that clears storage.
5. Host storage must be unlocked in the same way it was locked. If you issued the LOCK command with the LOGICAL operand to lock a guest page to a host logical storage page, the corresponding real storage frame was also locked by CP. To release both the host logical storage page and the real storage frame, you must also specify the LOGICAL operand on the UNLOCK command.
6. A guest page can be locked only once in host logical storage and once in real storage. If you issued the same LOCK command multiple times, only one UNLOCK command is needed.
7. If you issue UNLOCK with the ASIT operand and specify the ASIT of a space owned by a user that has been relocated by the VMRELOCATE command prior to a subsequent reset clear function, the UNLOCK command is rejected and error message HCPCPW033E is issued. To avoid this situation, issue a command that initiates a reset clear function on the user on the relocation target system. Then obtain the new ASIT value by using the QUERY SPACES command or the ADRSPACE QUERY API from a program. Commands that result in a reset clear include:
  - SYSTEM CLEAR
  - IPL by NSS name
  - IPL by device with the CLeAr option
  - SET MACHine (to a different machine architecture)

## Responses

Response 1:

```
Command complete
```

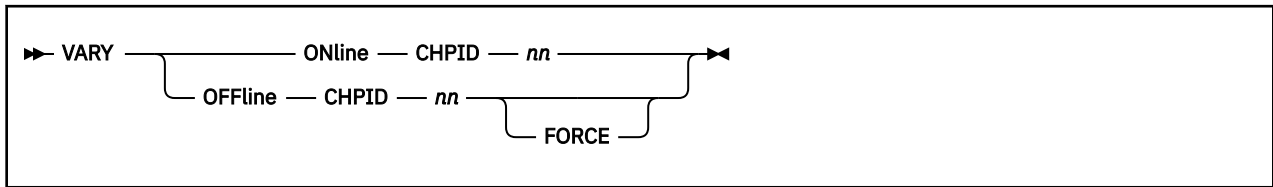
is issued when the command finished successfully. This response is displayed even if you did not have any pages locked in host storage.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}

- HCP004E Invalid hexloc - *hexloc*
- HCP005E Invalid hexdata - *hexdata*
- HCP009E Invalid range - *range*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP033E Hexloc missing or invalid
- HCP045E *userid* not logged on
- HCP160E Hexloc *hexloc* exceeds storage
- HCP410E CP ENTERED; PAGING ERROR
- HCP1003E *module* was not found.
- HCP6720I No items found to satisfy selection criteria.

## VARY CHPID



### Authorization

Privilege Class: B

### Purpose

Use VARY CHPID to enable or disable a real connection between the host system and one or more real devices.



**Attention:** The VARY ON CHPID command should be used instead of the corresponding function at the SE Hardware Console. Using the SE Hardware Console may cause channel path-type changes or channel path availability to be unrecognized by VM.

### Operands

#### ONline

tells CP to enable the specified channel path.

#### OFFline

tells CP to disable the specified channel path.

#### CHPID *nn*

identifies the path that is to become enabled (physically available) or disabled (unavailable) for use by the system to communicate with the specified device or devices.

#### FORCE

specifies that the path be unconditionally disabled for use by the system.

### Usage Notes

1. To successfully change the availability status of a CHPID, the CHPID must already have been defined to the system.
2. In order to take a real CHPID offline in an orderly fashion, you must disable the software's use of that CHPID. At the very least, this means you will need to vary the logical path offline (VARY OFF PATH). If the CHPID to be removed represents the last path to a device, then you will also need to vary the device offline (VARY OFF rdev) as well.
3. The FORCE operand will cause the REAL path (CHPID) to become unavailable in the hardware without regard for any software dependencies that may exist. In other words, if you choose to use the FORCE operand, the CHPID will be brought offline immediately without regard to any active I/O on the channel path. This operand is analogous to making the CHPID offline by using the hardware system console. This FORCE operand is not recommended for general use and should only be considered in such situations that are deemed critical to system survival.
4. If a CHPID is defined which has control units and devices defined to it (and these are all powered up), CP will immediately drive all the devices through device initialization when you issue the VARY ON CHPID command. For a device that is already online, CP will logically add the path to the device. Devices that are found to be offline will come online as long as CP can obtain all the needed device information.

5. If a CHPID is defined which has no control units or devices defined to it, then only the CHPID is brought online when you issue the VARY ON CHPID command. If you then define a control unit and devices to the CHPID, they will only come online if you issue the VARY ON *rdev* command.

## Responses

Response 1:

To enable CHPID 21, enter the following VARY CHPID command:

```
vary online chpid 21
Channel path 21 was successfully varied online
```

Response 2:

If the CHPID was already online:

```
vary online chpid 21
Channel path 21 is already online
Ready;
```

Response 3:

To disable CHPID 21, enter the following VARY CHPID command:

```
vary offline chpid 21
Channel path 21 was successfully varied offline
Ready;
```

Response 4:

If the CHPID was already offline:

```
vary offline chpid 21
Channel path 21 is already offline
Ready;
```

Response 5:

To disable CHPID 21 unconditionally, enter the following VARY CHPID command:

```
vary offline chpid 21 force
Channel path 21 was successfully varied offline
Ready;
```

## Messages

- HCP0003E Invalid option - command contains extra option(s) starting with *option*
- HCP0026E Operand missing or invalid
- HCP0846E Invalid channel path identifier.
- HCP6811E Vary {online | offline} CHPID *nn* failed with return code *nnnn*.
- HCP6814E Command is not valid on this processor.
- HCP6818E Vary command failed. Channel path *nn* is logically online.

## VARY CORE



### Authorization

Privilege Class: B

### Purpose

Use VARY CORE to add a processor or core to the configuration or remove it from the configuration. The status of multithreading determines whether the VARY CORE command will affect a processor or a core. The term *core* refers to a physical CPU resource. In a logical partition (LPAR), logical CPUs (processors) are dispatched on cores when multithreading is not installed or not enabled. However, when multithreading is enabled, the core might contain multiple threads, and logical CPUs are dispatched on the threads of the core.

When multithreading is not installed or not enabled, the terms processor and core are synonymous, and the VARY CORE command configures or deconfigures processors. However, when multithreading is enabled, only whole cores can be configured or deconfigured. You cannot configure or deconfigure individual processors. When a core is varied online, the number of threads per core that are supported by the hardware, the z/VM system, and the current multithreading settings are made available for use. The multithreading settings are set by the MULTITHREADING configuration statement and may be modified by the SET MULTITHREAD command. When a core is varied offline, all of the threads online on that core are stopped and removed.

### Operands

#### ONline

when multithreading is not installed or not enabled, adds a processor to the configuration and makes it available for use by CP. When multithreading is enabled, this operand adds a core to the configuration and makes its processors (threads) available for use by CP.

#### OFFline

when multithreading is not installed or not enabled, stops the use of a processor and removes it from the configuration. When multithreading is enabled, this operand stops the use of a core and removes its processors (threads) from the configuration.

#### CORE *nnnn*

specifies the processor or core to be made available or unavailable. When multithreading is not installed or not enabled, the value *nnnn* is a hexadecimal number (0000 - 004F) that identifies the address of the processor. When multithreading is enabled, the value *nnnn* is a 1- to 4-digit hexadecimal number that identifies the core ID.

### Usage Notes

1. Before using the VARY CORE command when multithreading is enabled, it might be useful to issue the QUERY PROCESSORS command to display a list of processors and their associated core IDs. For an example of how the core ID corresponds to a processor address, see [“QUERY MULTITHREAD” on page 1026](#).
2. The last active primary processor or core cannot be varied offline. A primary processor or core is one whose CPU type matches the CPU type of the system's IPL processor.



3. When multithreading is not installed or not enabled, if the last unparked processor of a secondary processor type is varied offline while all other online processors of that type are parked, CPU affinity will be temporarily suppressed until one of those processors can be unparked.

When multithreading is enabled, if the last core that contains unparked processors of a secondary processor type is varied offline while all other online processors of that type are parked, CPU affinity will be temporarily suppressed until one of those processors can be unparked.

A secondary processor is a processor whose CPU type does not match the CPU type of the system's IPL processor.

4. If the last processor of a secondary processor type is varied offline, CPU affinity will be suppressed for all user IDs and resource pools defined on the system that have CPU affinity on for that secondary processor type. As soon as a secondary processor of that type is varied online, CPU affinity will be unsuppressed. A resource pool that limits a secondary processor type (like IFL) and that has CPU affinity suppressed will apply the resource pool limit against the number of CP CPUs on the system. For example, if the resource pool limit is set to CAPACITY 1.5, then the resource pool is limited to 1.5 CP CPUs when CPU affinity is suppressed. If the resource pool limit is set to LIMITHARD 20%, then the resource pool is limited to 20% of the CP CPU resource when CPU affinity is suppressed and 20% of the IFL CPU resource when CPU affinity is unsuppressed.

## Responses

Response 1:

```
Command accepted
Core nnnn online Proc pppp[-pppp]
```

indicates that the specified core and its associated processor or processors are now online and operational.

Response 2:

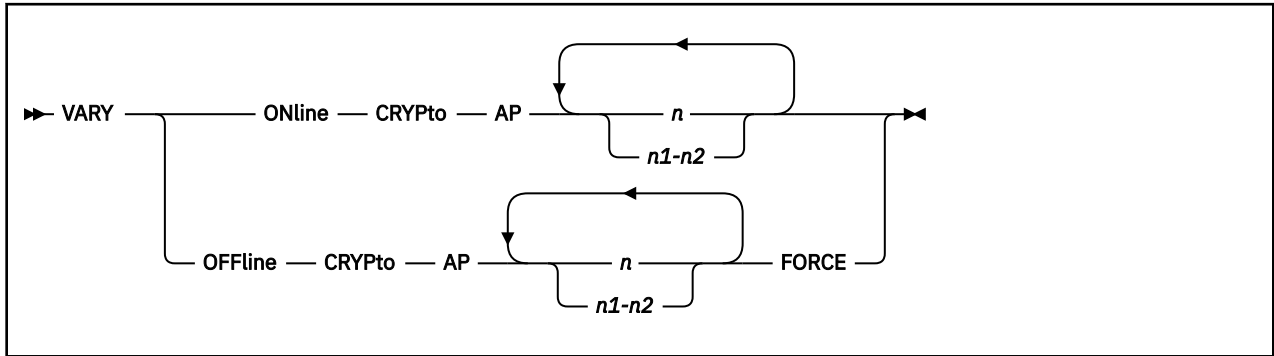
```
Command accepted
Core nnnn offline Proc pppp[-pppp]
```

indicates that the specified core and its associated processor or processors are now offline and not operational.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP039E Core ID missing or invalid
- HCP880E The command processing cannot complete because core *nnnn* is already online.
- HCP881E The command processing cannot complete because core *nnnn* is already offline.
- HCP882E The VARY command processing cannot complete due to an initialization failure.
- HCP883E The command processing cannot complete because processor *pppp* [within core *nnnn*] is not responsive.
- HCP884E The command processing cannot complete because core *nnnn* is the only active primary core.
- HCP885E The VARY processing cannot complete because the physical vary failed; core *nnnn* is logically offline.
- HCP886E The VARY command processing cannot complete because core *nnnn* is not available.
- HCP1463I CPU affinity is suppressed for the specified CPU type *type*
- HCP1465I CPU affinity is no longer suppressed for the specified CPU type *type*

## VARY CRYPTO



### Authorization

Privilege Class: B

### Purpose

Use VARY ONLINE CRYPTO to bring crypto adapters online and make them available for use.

Use VARY OFFLINE CRYPTO to take crypto adapters offline and make them unavailable for use.

If a crypto adapter is varied offline while resources on the adapter are attached to a virtual machine for dedicated use, or while resources are attached to the system for shared use, then the assignments remain intact while the adapter is offline. When the adapter is brought back online, the resources will be re-attached to the virtual machines or to the system as they were prior to the vary offline.

The FORCE option is required when the last functioning and usable resource in the shared pool is varied offline.

The FORCE option is required when the last functioning and usable resource that is dedicated to a virtual machine is varied offline.

### Operands

#### ONline

brings a crypto adapter online and makes it available for use.

#### OFFline

brings a crypto adapter offline and makes it unavailable for use.

#### *n*

#### *n1-n2*

specifies a single adapter number, a list of adapter numbers, a range of adapter numbers or a combination of these, to be varied online or offline. If you are specifying a list of numbers, insert at least one blank between the numbers you specify. If you are specifying a range of numbers, insert a hyphen (-) between the numbers you specify. The number is a decimal number between 0 and 255.

#### FORCE

The FORCE option is required when the last functioning and usable resource in the shared pool is varied offline.

The FORCE option is required when the last functioning and usable resource that is dedicated to a virtual machine is varied offline.

### Usage Notes

1. If an adapter is being brought online for the first time, VARY ONLINE CRYPTO should be followed by one of the following to make the specific resources available to a virtual machine or to the system:

- ATTACH CRYPTO
- Logon a virtual machine with a CRYPTO APDEDICATED statement in the directory entry.

If the adapter was previously online, existing resource assignments remain intact when it is brought back online, and the resources are again available for these virtual machines.

2. VARY OFFLINE CRYPTO with the FORCE option will cause the adapter to be taken offline even if it is the last resource attached to the system for shared use, or if it is the last resource attached to a virtual machine for dedicated use.
3. If CP is unable to take the adapter online or offline at the hardware level, the adapter will become logically online or offline. The operational status of the resources on the adapter will be reported in the Q CRYPTO response. A logically online resource could be deconfigured or in an error state, which would make it unusable. A logically offline resource could be operational, which would make it usable if it is dedicated to a virtual machine. CP will not use offline crypto resources that are in the shared pool.
4. For information on planning and managing crypto resources on a z/VM system, see [Crypto Planning and Management in z/VM: CP Planning and Administration](#).

## Responses

Response 1:

This response indicates that the specified adapter is now online.

```
CRYPTO AP apnum VARIED ONLINE
CRYPTO AP apnum VARIED ONLINE by userid
```

Response 2:

This response indicates that the specified adapter is now offline.

```
CRYPTO AP apnum VARIED OFFLINE
CRYPTO AP apnum VARIED OFFLINE by userid
```

Response 3:

This response is sent to the primary system operator when the command issuer is not the system operator.

```
CRYPTO AP apnum VARIED ONLINE by userid
CRYPTO AP apnum VARIED OFFLINE by userid
```

### ***apnum***

is the crypto adapter number.

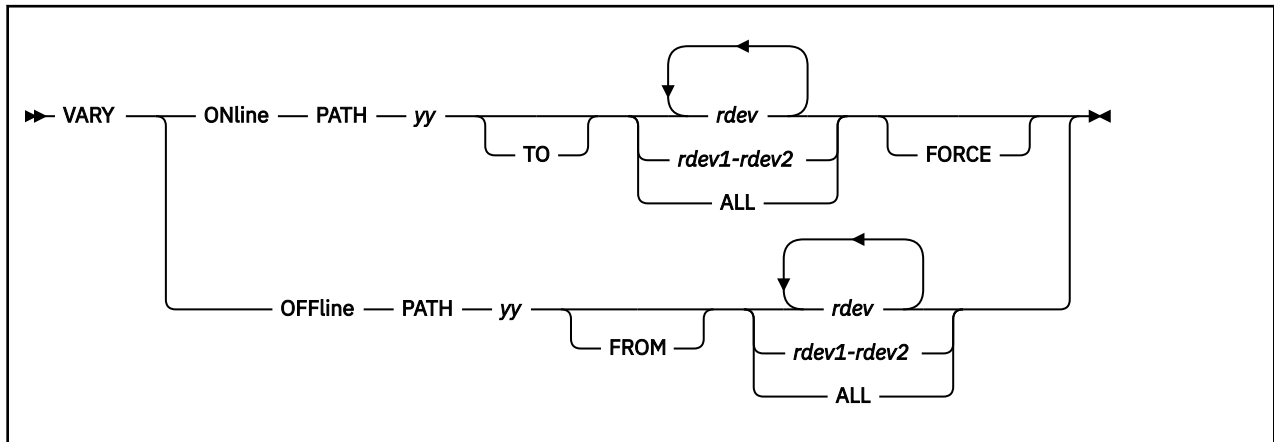
## Messages

- HCP003E Invalid option - *option*
- HCP003E Invalid option - *option*| command contains extra option(s) starting with *option*
- HCP1014E A required option is missing
- HCP1725E The VARY command cannot complete because crypto AP *apnum* is already online.
- HCP1725E The VARY command cannot complete because crypto AP *apnum* is already offline.
- HCP1741E The VARY command cannot complete because crypto AP *apnum* has the last operational resource attached to *domnum*. The FORCE option is required.
- HCP1741E The VARY command cannot complete because crypto AP *apnum* has the last operational resource attached to SYSTEM. The FORCE option is required.
- HCP1742E AP The VARY CRYPTO command could not complete because crypto AP *apnum* is not assigned to this system.

## VARY CRYPTO

- HCP1743E The VARY command cannot complete because the physical vary failed. AP *apnum* is logically online.
- HCP1743E The VARY command cannot complete because the physical vary failed. AP *apnum* is logically offline.
- HCP2768E Missing crypto AP number
- HCP6706E Invalid Crypto AP number

## VARY PATH



### Authorization

Privilege Class: B

### Purpose

Use VARY PATH to enable or disable a logical connection between the host system and one or more real devices.

### Operands

#### ONline

#### OFFline

enables or disables the logical connection.

#### yy

identifies the path to become logically available or logically unavailable for use by the system to communicate with the specified device or devices.

**Note:** If the specified path is the last available path to the specified device, CP attempts to vary the device offline. If the device cannot be varied offline, the VARY OFFLINE PATH request is denied.

#### rdev

#### rdev1-rdev2

is the real device number of the devices to which the specified path is made logically available or logically unavailable. You may specify a single device, a list, or a range of devices. If you specify a range of real device numbers, all the valid devices in the range, including the first and last devices you specified, will be processed. The maximum number of devices allowed in a range is 256. For a device in the active configuration, the device number entered can be a 4-digit hexadecimal number between X'0000' and X'FFFF'. For a device in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured. A range of devices cannot span subchannel sets.

#### ALL

specifies that the path be made logically available or logically unavailable to all real devices that have access to it.

#### FORCE

specifies that an attempt be made to bring the path online even if another function had previously varied it offline.

## Usage Notes

1. The VARY PATH command does not start the service processor or cause any hardware functions to be performed.
2. If you specify a list or range of devices, an attempt is made to vary the specified path online or offline to each device. An error message is issued for every device for which the specified path cannot be varied online or offline. Check examples after these usage notes.
3. Paths leading to dedicated non-dynamic pathing devices that allow RESERVE/RELEASE are not varied offline.  
 To determine if RESERVE/RELEASE is allowable, enter the QUERY DASD command and see if the response from the device indicates that the device is SHARED, RESERVED, or has a RESERVE PENDING (see the Responses section of the QUERY DASD command).
4. Inoperational paths may be varied online because no I/O is issued to validate the path when a non-dynamic pathing device is reserved or is dedicated to a guest and supports RESERVE/RELEASE.
5. When varying a path offline:
  - Vary the path offline from the guest operating system if the guest (such as MVS/ESA) supports a VARY PATH command.
  - Vary the path offline from the host system (z/VM).
6. When varying a path online:
  - Make the path physically available.
  - Vary the path online to the host (z/VM).
  - Vary the path online to the guest operating system if the guest supports a VARY PATH command.
7. If the IBM Service Representative indicates that the service on the path is complete but the control unit initiated resume did not successfully vary the affected path online, the FORCE option may be used to vary the quiesced path back online.

## Examples

Enter vary on path c2 to 140 to make path C2 leading to device 140 available to that device. A typical response may appear as follows:

```
Vary path c2 online command initiated
Vary path c2 online command complete
```

Enter vary off path c2 from 0190-0197 to make path C2 unavailable to devices 0190 through 0197. A typical response may appear as follows:

```
Vary path c2 offline command initiated
Path c2 not varied offline from device 0192
because it is the last path to device
Vary path c2 offline command complete
```

## Responses

Response 1: Vary Initiated:

```
VARY PATH yy {ONLINE } COMMAND INITIATED
           {OFFLINE}
```

indicates that the request has been initiated.

**yy**

is the path requested to be varied online or offline.

Response 2: Vary Complete:

```
VARY PATH yy {ONLINE } COMMAND COMPLETE
             {OFFLINE}
```

indicates that command processing is complete. An attempt was made to make the specified path available or unavailable (whichever was specified—online or offline) to each device specified. While command processing is now complete, check carefully any console messages that might indicate that not all processing was successful.

***yy***

is the path requested to be varied online or offline.

Response 3: Path Not Operational:

```
PATH yy NOT VARIED ONLINE
TO   DEVICE rdev; PATH NOT OPERATIONAL
```

indicates that the specified path has not been made available (varied online) for the specified device because the path is not operational. Make the path operational and then retry the command.

***yy***

is the path not made operational.

***rdev***

is the real device number.

Response 4: Last Path to Device:

```
PATH yy NOT VARIED OFFLINE FROM DEVICE rdev
BECAUSE IT IS THE LAST PATH TO DEVICE
```

indicates that the specified logical path is the last path to the specified device. CP tried unsuccessfully to vary the device offline. Use the VARY command to vary the device offline, then retry the VARY PATH command.

***yy***

is the path CP tried to make unavailable (vary offline) to the specified device.

***rdev***

is the real device number.

Response 5: Path Not Installed To Device:

```
PATH yy NOT VARIED {ONLINE TO } DEVICE rdev;
                   {OFFLINE FROM}
```

Continued:

```
PATH DOES NOT EXIST TO THE DEVICE
```

indicates that the specified path was not installed for the specified device.

***yy***

is the path requested to be made logically available (varied online) or unavailable (varied offline) to the specified device.

***rdev***

is the real device number.

Response 6: Device Reserve Held:

```
PATH yy NOT VARIED OFFLINE FROM DEVICE rdev;
RESERVED PATH
```

indicates that the specified path was not taken offline because there is a reserved allegiance to that path, which could not be moved to another path. (See Usage Note “3” on page 2054.)

***yy***

is the number for the path not taken offline.

## VARY PATH

### **rdev**

is the real device number.

Response 7: First Path:

```
FIRST PATH yy TO DEVICE rdev  
VARIED ONLINE; DEVICE VARIED ONLINE
```

indicates that the device was previously offline because all of its channel paths were varied offline. The path requested to be varied online was the first path to the device to be varied online, and therefore the device as well as the path was varied online.

### **yy**

is the path that was requested to be made available (varied online) to the specified device.

### **rdev**

is the real device number.

Response 8: Last Path:

```
LAST PATH yy TO DEVICE rdev  
VARIED OFFLINE; DEVICE VARIED OFFLINE
```

indicates that the path requested to be varied offline was the last path to the device and therefore the device (and the path) were varied offline.

### **yy**

is the path that was requested to be made unavailable (varied offline) to the specified device.

### **rdev**

is the real device number.

Response 9: Pending Reserve:

```
PATH yy NOT VARIED OFFLINE FROM  
DEVICE rdev; DEVICE RESERVE PENDING
```

indicates that the path requested to be varied offline could not be varied offline because a reserve request was pending to that device. Check the other system sharing this device; it may be hung, holding the reserve, thus prohibiting CP's reserve request from being granted.

### **yy**

is the number of the path not varied offline because of a pending reserve request.

### **rdev**

is the real device number.

Response 10: Path Not Physically Available:

```
PATH yy NOT VARIED ONLINE TO DEVICE rdev;  
PATH NOT AVAILABLE TO THE DEVICE
```

indicates that the path is not physically available to the device. Ensure that the path is switched in and varied online from the system's service console.

### **yy**

is the number of the path not physically available to the device.

### **rdev**

is the real device number.

Response 11: Path Already Varied Offline:

```
PATH yy HAS ALREADY BEEN VARIED OFFLINE  
FOR DEVICE rdev
```

indicates that the system operator has already varied the path offline.



***yy***

is the path that was both requested to be and then varied offline.

***rdev***

is the real device number.

Response 12: Path Already Online:

```
PATH yy ALREADY ONLINE FOR DEVICE rdev
```

indicates that the path is already online.

***yy***

is the number of the path already online.

***rdev***

is the real device number.

Response 13: Configuration Change Detected:

```
PATH yy NOT VARIED ONLINE TO DEVICE rdev;  
A CONFIGURATION CHANGE WAS DETECTED.
```

indicates that a configuration change was detected.

***yy***

is the number of the path not varied online.

***rdev***

is the real device number.

Response 14: Path Offline By Configuration Manager:

```
PATH yy NOT VARIED ONLINE TO DEVICE rdev;  
PATH OFFLINE BY CONFIGURATION MANAGER
```

indicates that the path is not varied online because the configuration manager has varied it offline.



**CAUTION:** If it is necessary to bring the path online, reissue the VARY PATH command with the FORCE option.

***yy***

is the channel path identifier.

***rdev***

is the real device number.

Response 15: Path Offline Due to a Control Unit Initiated Request:

```
PATH yy NOT VARIED ONLINE TO DEVICE rdev  
BECAUSE IT IS QUIESCED FOR MAINTENANCE.
```

indicates that the path is not varied online because a control unit initiated quiesce request has varied it offline. If an IBM Service Representative has indicated that the service on the path is complete and the control unit initiated resume request has failed to successfully vary the affected path online, the FORCE option of the VARY ONLINE PATH command may be used to vary the quiesced path back online.

***yy***

is the channel path identifier.

***rdev***

is the real device number.

Response 16: Subchannel Disabled:

```
PATH yy NOT VARIED ONLINE TO DEVICE rdev  
SUBCHANNEL IS DISABLED.
```

## VARY PATH

indicates that the path was not varied online because the associated subchannel for this device is disabled (offline). The subchannel needs to be varied online to be enabled. Once the subchannel is enabled, the path can be varied online.

***yy***

is the channel path identifier.

***rdev***

is the real device number.

Response 17: Device is BOXED:

```
PATH yy NOT VARIED ONLINE TO DEVICE rdev
Device is BOXED.
```

indicates that the path has not been made available (varied online) to the device because the device is BOXED.

If the path specified was marked 'offline by an authorized user' (as a result of a previous VARY OFF PATH command), then this path will no longer be marked that way. If a previous VARY ON DEVICE command failed because no paths were available to this device, then it may have been due to all paths to the device being in the 'offline by authorized user' state. If so, then a subsequent VARY ON DEVICE command may now be successful because it can use this path.

***yy***

is the number of the path not varied online because the device is BOXED.

***rdev***

is the real device number.

Response 18: Incompatible High Performance FICON feature:

```
PATH yy NOT VARIED ONLINE TO DEVICE rdev; path offline
due to incompatible High Performance FICON feature.
```

indicates that incompatible paths cannot be varied online logically unless the FORCE option is specified.

***yy***

is the number of the path not varied online.

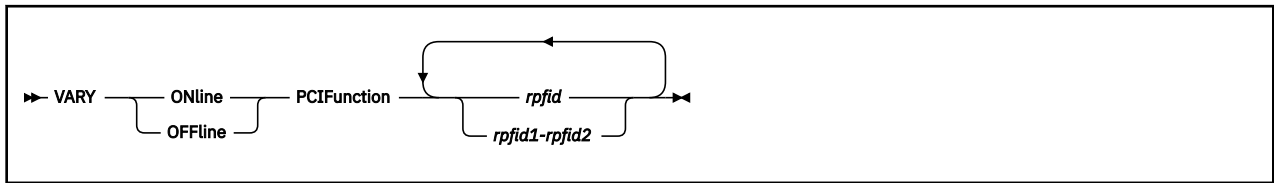
***rdev***

is the real device number.

## Messages

- HCP003E Invalid option - *{option|command}* contains extra option(s) starting with *option*}
- HCP009E Invalid range - *range*
- HCP021E A real device number was not supplied or it is invalid.
- HCP026E Operand missing or invalid
- HCP040E Device *{rdev/vdev/ldev}* does not exist
- HCP846E Invalid channel path identifier.
- HCP1150E DASD *{rdev}* is not a valid base exposure.
- HCP1280E Device *vdev* not usable; reserve lost [by *userid* as *vdev*]
- HCP1281I Path *channel* to device *device* now offline; path groupid mismatch.
- HCP1282I Path *{yy}* to device *{rdev}* has been forced offline; a configuration change was detected.
- HCP1550E *value* is an invalid device class.
- HCP6000E The range of device numbers cannot exceed 256.

## VARY PCIFUNCTION



### Authorization

Privilege Class: B

### Purpose

Use VARY PCIFUNCTION to configure or deconfigure a PCI function.

### Operands

#### ONline

configures a PCI function.

#### OFFline

deconfigures a PCI function.

#### PCIFunction

used for specifying PCI function(s).

#### *rpfid*

#### *rpfid1-rpfid2*

specifies a real PCI function ID or a range of PCI function IDs that are to be enabled or disabled for the user. Specify a PCI function ID as a 1- to 8-digit hexadecimal number between X'00000000' and X'7FFFFFFF'. The total number of PCI functions specified cannot exceed 256.

### Responses

Response 1:

```
rpfid varied {online }
             {offline}
```

indicates that the specified real PCI function ID is now online or offline.

Response 2:

```
rpfid already {online }
              {offline}
```

indicates that the specified real PCI function ID is already online or offline.

Response 3:

```
nnn PCI function(s) specified; nnn PCI function(s) successfully varied {online }
                                                                           {offline}
```

is a summary response indicating how many PCI functions were successfully varied online or offline.

Response 4:

```
PCI function rpfid not varied online; PCI function could not be configured.
```

indicates that the PCI function specified to be varied online could not be configured.

Response 5:

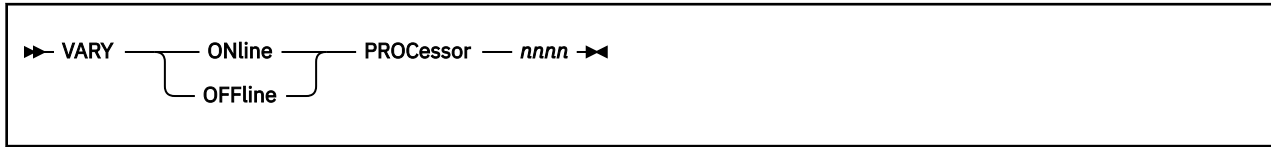
```
PCI function rpfid not varied offline; PCI function could not be
deconfigured.
```

indicates that the PCI function specified to be varied offline could not be deconfigured.

### Messages

- HCP002E Invalid operand - *operand*
- HCP009E Invalid range - *range*
- HCP021E A real PCI function ID was not supplied or it is invalid.
- HCP040E PCI function {*rpfid*} does not exist
- HCP040E PCI function {*rpfid*} attached to *userid*.
- HCP1163I PCI function *rpfid* cannot be allowed online because PCI support is not enabled for the system.
- HCP1163I PCI function *rpfid* initialization has failed; return code *rc*, response code *respcode*.
- HCP1167E PCI function *rpfid* cannot be brought online because it is of an unknown type.
- HCP6000E The range of PCI function IDs cannot exceed 256.
- HCP6790I VARY processing halted. The next function that would have been processed is *rpfid*

## VARY PROCESSOR



### Authorization

Privilege Class: B

### Purpose

When multithreading is not installed or not enabled, use VARY PROCESSOR to add a processor to the configuration for use by CP or to remove a processor from the configuration. When multithreading is enabled, the VARY PROCESSOR command is not valid and the VARY CORE command must be used.

### Operands

#### ONline

adds a processor to the configuration and makes it available for use by CP.

#### OFFline

stops the use of a processor and removes it from the configuration.

#### PROcEssor *nnnn*

specifies the processor to be made available or unavailable. The value *nnnn* is a hexadecimal number (0000 - 004F) that identifies the address of the processor.

### Usage Notes

1. The last active primary processor cannot be varied offline. A primary processor is a processor whose CPU type matches the type of the system's IPL processor.
2. If the last unparked processor of a secondary processor type is varied offline while all other online processors of that type are parked, CPU affinity will be temporarily suppressed until one of those processors can be unparked.  
  
A secondary processor is a processor whose CPU type does not match the CPU type of the system's IPL processor.
3. If the last processor of a secondary processor type is varied offline, CPU affinity will be suppressed for all user IDs and resource pools defined on the system that have CPU affinity on for that secondary processor type. As soon as a secondary processor of that type is varied online, CPU affinity will be unsuppressed. A resource pool that limits a secondary processor type (like IFL) and that has CPU affinity suppressed will apply the resource pool limit against the number of CP CPUs on the system. For example, if the resource pool limit is set to CAPACITY 1.5, then the resource pool is limited to 1.5 CP CPUs when CPU affinity is suppressed. If the resource pool limit is set to LIMITHARD 20%, then the resource pool is limited to 20% of the CP CPU resource when CPU affinity is suppressed and 20% of the IFL CPU resource when CPU affinity is unsuppressed.

### Responses

Response 1:

```
PROCESSOR nnnn ONLINE
```

indicates that the specified processor is now online and operational.

Response 2:

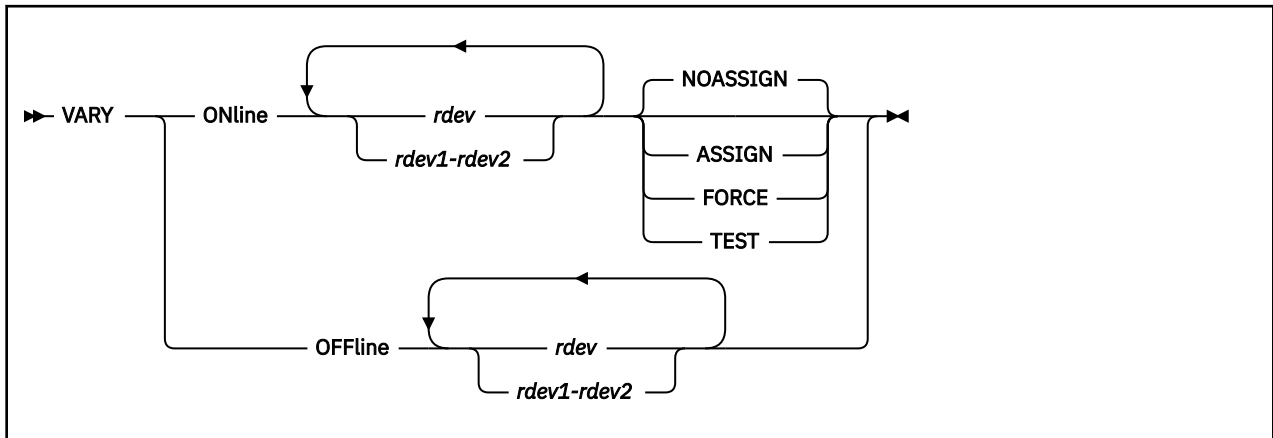
```
PROCESSOR nnnn OFFLINE
```

indicates that the specified processor is now offline and not operational.

### Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP026E Operand missing or invalid
- HCP039E Processor address missing or invalid
- HCP880E The command processing cannot complete because processor *nnnn* is already online.
- HCP881E The command processing cannot complete because processor *nnnn* is already offline.
- HCP882E The VARY command processing cannot complete due to an initialization failure.
- HCP883E The command processing cannot complete because processor *nnnn* is not responsive.
- HCP884E The command processing cannot complete because processor *nnnn* is the only active primary processor.
- HCP885E The VARY processing cannot complete because the physical vary failed; processor *nnnn* is logically offline.
- HCP886E The VARY command processing cannot complete because processor *nnnn* is not available.
- HCP1321E VARY PROCESSOR is not valid because multithreading is enabled.
- HCP1463I CPU affinity is suppressed for the specified CPU type *type*
- HCP1465I CPU affinity is no longer suppressed for the specified CPU type *type*

## VARY (Real Device)



### Authorization

Privilege Class: B

### Purpose

Use VARY (real device) to do the following:

- Enable or disable a device for the control program (CP)
- Enable or disable a device for the user.

### Operands

#### ONline

#### OFFline

enables or disables the logical connection.

#### *rdev*

#### *rdev1-rdev2*

is the number of the real device to be varied online or offline. You may specify a list or range of devices. If you specify a range of real device numbers, all valid devices in the range, including the first and last devices you specify, are varied online or offline. For a device in the active configuration, the device number entered can be a 4-digit hexadecimal number between X'0000' and X'FFFF'. For a device in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured. A range of devices cannot span subchannel sets.

#### NOASSIGN

#### ASSIGN

tells CP to try to ASSIGN or UNASSIGN the device to CP at vary on time. If the ASSIGN fails, the device is not varied online.

#### FORCE

specifies that an attempt be made to bring the device online even if another function has prevented the device from coming online.

#### TEST

specifies that CP bring the device online even if conditions exist that would normally prevent the device from successfully being brought online.

## Usage Notes

1. When a device is being brought online, the system will attempt to generate a device equivalency ID (EQID). However, a user-defined EQID might have already been assigned to the device. In this case, if the system successfully generates an EQID, it will override whatever EQID the user had assigned the device, and you will receive message HCP6627I. Note that this does not apply to the following device types: CTCA, FCP, HiperSockets, and OSA.
2. If you specify a list or range of devices, an attempt is made to vary them online or offline regardless of whether an error is encountered with any one of the specified devices. An error message is issued for every device that cannot be varied online or offline. A summary message is issued to indicate the number of devices successfully varied online or offline.
3. An UNASSIGN is issued to all tape devices that support ASSIGN/UNASSIGN during the vary off process.
4. If the IBM Service Representative indicates that the service on a quiesced device or set of devices is complete, but the control unit initiated resume did not successfully vary the affected device(s) online, the FORCE operand may be used to vary the device(s) back online.
5. The TEST operand provides the ability for a device that would not normally be allowed online by CP, to be brought online in *test mode* so that it can be attached to a virtual machine for the execution of diagnostics in problem determination. For a device to be eligible for *test mode*, it must be predefined either in the system configuration file or through a SET RDEVICE, as an unsupported device.

## Responses

Response 1:

```
rdev VARIED {ONLINE }
             {OFFLINE}
```

indicates that the specified device is now online or offline.

Response 2:

```
type rdev IS ALREADY {ONLINE }
                    {OFFLINE}
```

indicates that the specified device is already online or offline.

Response 3:

```
nnn DEVICE(S) SPECIFIED; nnn
DEVICE(S) SUCCESSFULLY VARIED {ONLINE }
                               {OFFLINE}
```

is a summary response indicating how many devices were successfully varied online or offline.

Response 4:

```
type rdev CANNOT BE VARIED ONLINE;
DEVICE HAS NO ONLINE CHANNEL PATHS
```

indicates that the device specified to be varied online has no online (logically available) channel paths. A previous vary path request has made the last path to the device unavailable causing the device to be varied offline.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP006E Invalid device type - *rdev*
- HCP009E Invalid range - *range*
- HCP021E A real device number was not supplied or it is invalid.

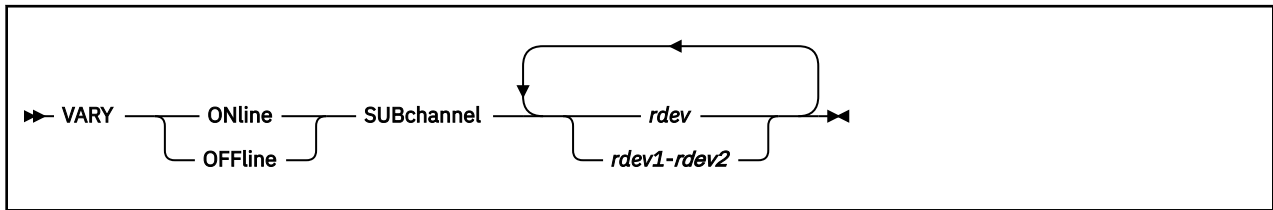


- HCP026E Operand missing or invalid
- HCP040E Device {*rdev/vdev/ldev*} does not exist
- HCP049E *Type rdev* in use
- HCP123E DASD *rdev* CP-owned
- HCP124E DASD *rdev* in use by *nnnnn* users
- HCP140E *Type rdev* attached to *userid*
- HCP142E *Type rdev* not drained
- HCP143E *Type rdev* in use by system
- HCP362I Device *rdev* cannot be varied online because the subchannel-id is invalid.
- HCP1150E DASD *rdev* is not a valid base exposure.
- HCP1250I Device *rdev* cannot be varied online because a time-out condition occurred.
- HCP1251I Device *rdev* cannot be varied online because it requires operator intervention.
- HCP1280E Device *vdev* not usable; reserve lost [by *userid* as *vdev*].
- HCP1281I Path *channel* to device *Device* now offline; path group ID mismatch.
- HCP1550E *Rdev* is an invalid device class.
- HCP1917E Tape device(s) {*rdev[-rdev]*|*vdev[-vdev]*} attached MULTIUSER.
- HCP3591E The hardware/software is not at the correct level to support LUN reset. Device *deviceFunction: function*
- HCP3593E *fp\_ioctl* for SCIOLCMD failed. resource name=*name*, return value=*rv*.Function: *function*
- HCP3594E *fp\_ioctl* for GET\_ROUTER\_STATE failed. resource name=*name*, return value=*rv*.Function: *function*
- HCP3595E *fp\_ioctl* for SET\_ROUTER\_STATE failed. resource name=*name*, return value=*rv*.Function: *function*
- HCP3597E *fp\_ioctl* for MODESENSE failed. resource name=*name*, return value=*rv*.Function: *function*
- HCP3598E *fp\_ioctl* for RTADDPATH failed. resource name=*name*, return value=*rv*.Function: *function*
- HCP3599E The cache settings for LUN *lun* will allow cached data to be lost if path failover occurs.
- HCP3603E Paths to disk *disk* do not go to same device.Function: *function*
- HCP6000E The range of device numbers cannot exceed 256.
- HCP6276I Device *rdev* cannot be varied online because the subchannel is not operational.
- HCP6277I Device *rdev* cannot be varied online because I/O errors occurred.
- HCP6280I Device *rdev* cannot be varied online because an error occurred while attempting to establish a path group.
- HCP6281I Device *rdev* cannot be varied online because of conflicting device identification information.
- HCP6282I Device *rdev* cannot be varied online because the subchannel is invalid.
- HCP6283I Device *rdev* cannot be varied online because no channel path is available.
- HCP6284I Device *rdev* cannot be varied online due to conflicting device classification information.
- HCP6285I Device *rdev* cannot be varied online due to conflicting device numbers.
- HCP6286I Device *rdev* cannot be varied online because the subchannel cannot be modified.
- HCP6287I Device *rdev* cannot be varied online because it did not respond to a no-op command.
- HCP6288I Device *rdev* cannot be varied online because the subchannel is varied offline
- HCP6289I Device *rdev* cannot be varied online because it is quiesced for maintenance.
- HCP6290I Device *rdev* cannot be varied online because it is uninstalled or unsupported.
- HCP6291I Device *rdev* cannot be varied online because it is assigned elsewhere.
- HCP6291I Device *rdev* cannot be varied online because it is reserved to another system.

## VARY (Real Device)

- HCP6292I Device *rdev* cannot be varied online because the ASSIGN option was specified and the device does not support assign.
- HCP6293I Device *rdev* cannot be varied online because it is a spare volume within the DASD subsystem.
- HCP6295I DASD *rdev* cannot be varied online because an associated DASD does not exist in the primary subchannel set.
- HCP6627I User-defined EQID replaced by system-generated identifier for device *rdev*.
- HCP6785I Unable to identify device *rdev* dynamically.
- HCP6861I Device *rdev* cannot be varied online because it is an alias Parallel Access Volume for which the base could not be located.
- HCP8700I Emulated Device *rdev* cannot be varied online because CP was unable to open the device.
- HCP8700I Emulated Device *rdev* cannot be varied online because its size is greater than 2,147,483,640 512-byte blocks.
- HCP8700I Emulated Device *rdev* cannot be varied online because there are no valid paths defined to the device.
- HCP8701I Path FCP\_DEV *rdev1* WWPN *wwpn* LUN *lun* was deleted from EDEV *rdev2* because it conflicts with online EDEV *rdev3*.
- HCP8701I Path FCP\_DEV *rdev1* WWPN *wwpn* LUN *lun* was deleted from EDEV *rdev2* because the FCP device is currently reserved for the EXPLORE command.
- HCP8701I FBA DASD *rdev* cannot be varied online because its size is greater than 2,147,483,640 512-byte blocks.

## VARY SUBCHANNEL



### Authorization

Privilege Class: B

### Purpose

Use VARY SUBCHANNEL to enable or disable one or more subchannels.

### Operands

#### ONline

tells CP to enable the specified subchannel, subchannels, or range of subchannels.

#### OFFline

tells CP to disable the specified subchannel, subchannels, or range of subchannels.

#### SUBchannel *rdev*

#### SUBchannel *rdev1-rdev2*

is the number of the real device whose associated subchannel is to be enabled or disabled. You may specify a list or range of devices. If you specify a range of real device numbers, all valid devices in the range, including the first and last devices you specify, will have their designated subchannels enabled or disabled. The maximum number of devices allowed in a range is 256. For a device in the active configuration, the device number entered can be a 4-digit hexadecimal number between X'0000' and X'FFFF'. For a device in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set where the device is configured. A range of devices cannot span subchannel sets.

### Usage Notes

1. Each RDEV has only one associated subchannel which defines this device to the hardware I/O configuration. Through this command CP will either set or reset the ENABLED bit in the associated subchannel for the given device.
2. Once you have successfully disabled the subchannel for the given device, no further I/O requests and I/O interrupts will be accepted for the device.
3. You must use the VARY command to take a device offline before you can disable its subchannel. Before you can VARY the device back online, you must enable the subchannel through the use of VARY SUBCHANNEL.

### Responses

Response 1:

To VARY a subchannel online, enter the following VARY SUBCHANNEL command:

```
vary on sub 701
0701 subchannel varied online
1 device(s) specified; 1 device(s) successfully varied subchannel online
Ready;
```

## VARY SUBCHANNEL

Response 2:

To VARY a range of subchannels offline, enter the following VARY SUBCHANNEL command:

```
vary off sub 701-703
0701 subchannel varied offline
0702 subchannel varied offline
0703 subchannel varied offline
3 device(s) specified; 3 device(s) successfully varied subchannel offline
Ready;
```

Response 3:

If the subchannel is already enabled:

```
vary on sub 702
0702 subchannel is already online
1 device(s) specified; 0 device(s) successfully varied subchannel online
Ready;
```

Response 4:

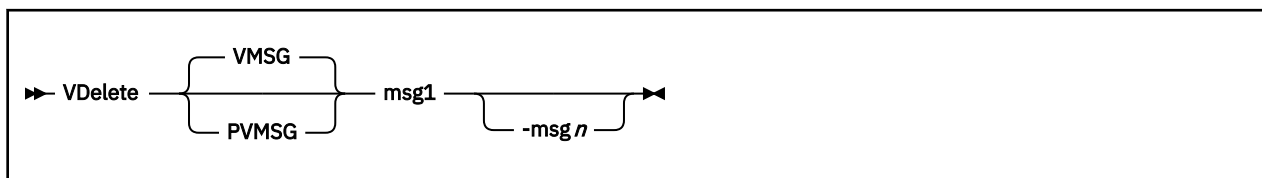
If the subchannel is already disabled:

```
vary off sub 701
0701 subchannel is already offline
1 device(s) specified; 0 device(s) successfully varied subchannel offline
Ready;
```

### Messages

- HCP009E Invalid range - *range*
- HCP0026E Operand missing or invalid
- HCP0040E Device *rdev* does not exist
- HCP0267E A subchannel number was not supplied or is invalid
- HCP1150E DASD *rdev* is not a valid base exposure.
- HCP1249I Device *rdev* cannot be varied subchannel {disable|enable} because a time-out condition occurred
- HCP6760E Device *rdev* is not offline.

## VDELETE



### Authorization

Privilege Class: G

### Purpose

Use VDELETE to delete selected messages from the list of messages generated by the QUERY VMSG/ PVMSG command and directed to your system console.

The z/VM control program, as part of its service processor simulation, intercepts system console messages issued by a guest operating system and displays them on the virtual machine operator console. Until an intercepted message is deleted, it remains on the z/VM message list and may be recalled by the virtual machine console operator.

Messages may be deleted in these ways:

- Your virtual machine operating system, through communication with the service processor, may delete it.
- The VDELETE command may be issued against the message.
- The message list is limited to 16 current entries; older messages are removed from the list.

### Operands

#### VMSG

indicates that nonpriority messages are to be deleted. The message numbers, accepted as parameters on this command, are displayed on the virtual machine operator console when QUERY VMSG is issued to recall outstanding nonpriority messages.

#### PVMSG

indicates that priority messages are to be deleted. The message numbers, accepted as parameters on this command, are displayed on the virtual machine operator console when QUERY PVMSG is issued to recall outstanding priority messages.

#### msg1 [-msgn ]

designates the numbers of the messages to be deleted. These numbers are displayed on the virtual machine operator console when QUERY VMSG/PVMSG is issued.

### Usage Notes

1. z/VM treats each line of multiple-line message text as a separate message when it is displayed on your operator console and saved for recall by using QUERY VMSG/PVMSG. However, when any individual line of a multi-line message is deleted, all of the multiple-line message is deleted.
2. The message numbers preceding the operating system message text are assigned by the QUERY VMSG/PVMSG command processor and QUERY must be entered before a message may be deleted using VDELETE.

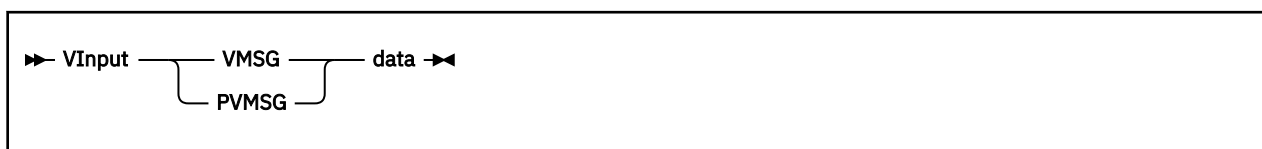
### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid

## VDELETE

- HCP6532I There is no {priority|non-priority} message currently outstanding with message number *nnnn* assigned to it.

## VINPUT



### Authorization

Privilege Class: G

### Purpose

Use VINPUT to request the z/VM control program to send the accompanying data to the operating system running in your virtual machine. To the operating system, the data appears to have been entered from the system console.

z/VM, as part of its service processor hardware simulation, assigns to your virtual machine operator console a subset of system console functions. If the operating system running in your virtual machine is able to receive commands from the system console, the specified command is sent by the control program to the operating system where it is interpreted and processed as though it had been entered from the system console.

### Operands

#### VMSG

requests that the command is to be sent as a nonpriority command.

#### PVMSG

requests the command is to be sent as a priority command.

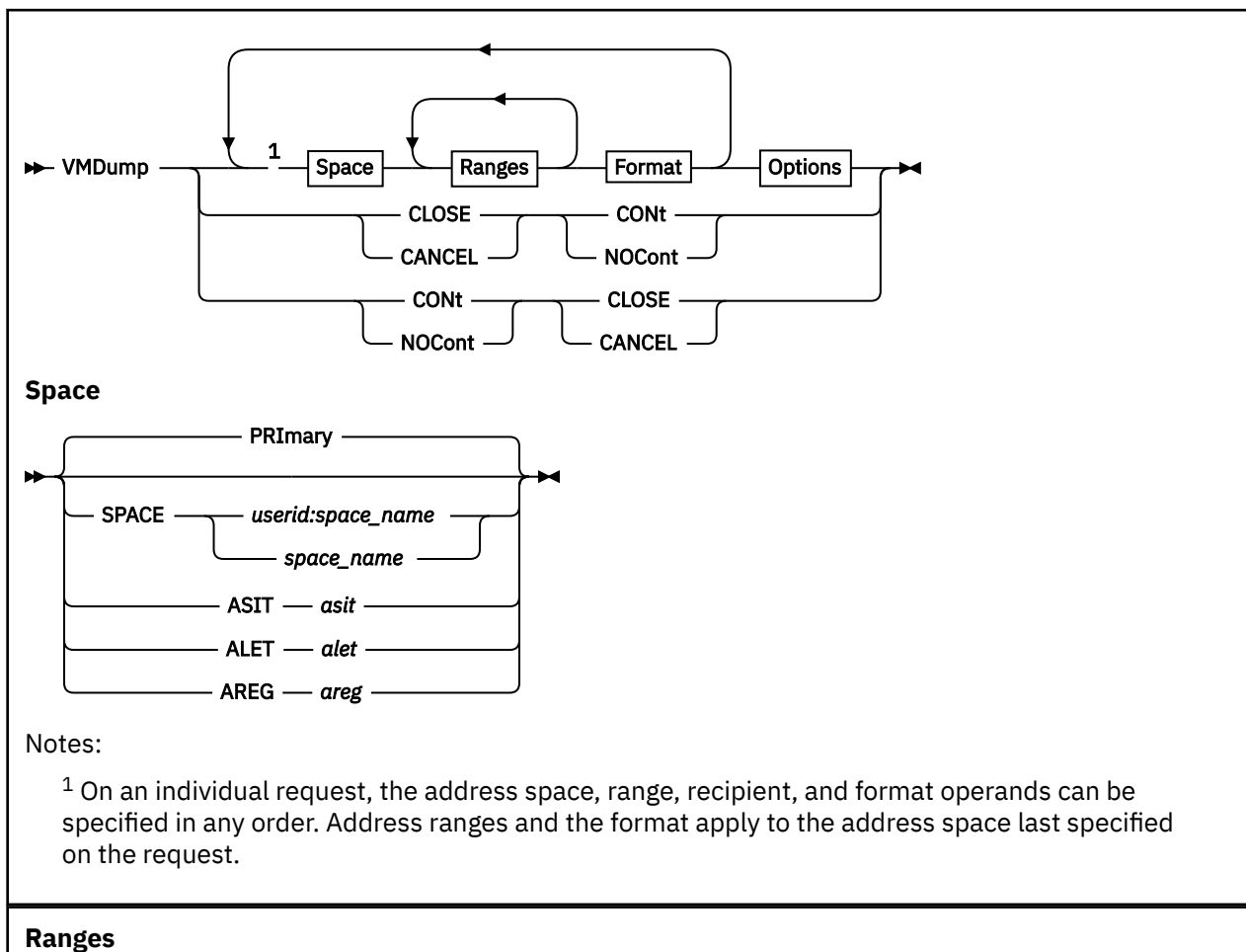
#### *data*

is the command to be sent to the operating system running in your virtual machine.

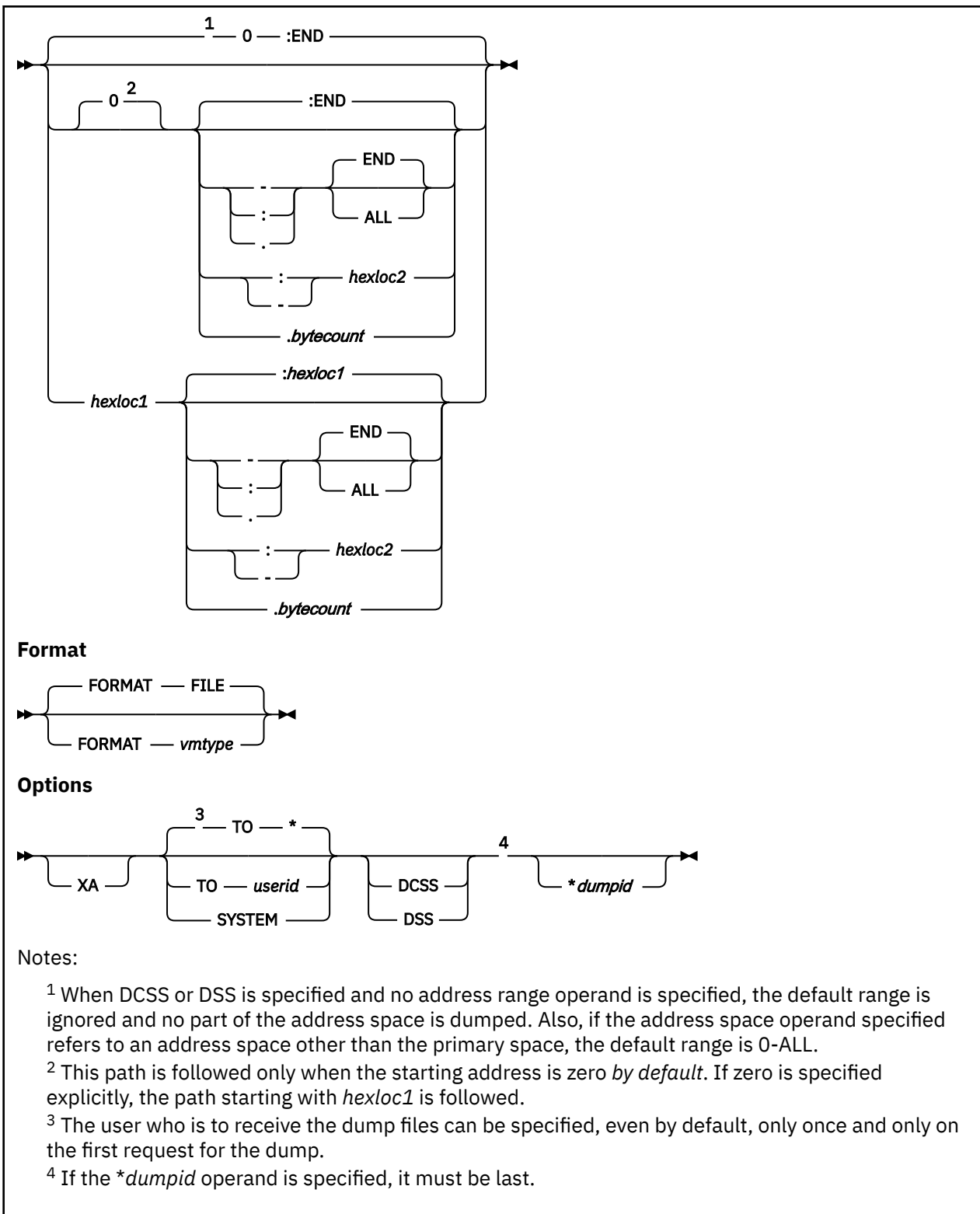
### Messages

- HCP026E Operand missing or invalid
- HCP6531E The operating system will not accept commands from the service processor.

## VMDUMP







### Authorization

Privilege Class: G

### Purpose

Use VMDUMP to dump all or selected pages from second-level storage.

## Operands

**Note:** An address space operand (PRIMARY, SPACE, ALET, ASIT, or AREG) qualifies the operands that follow it. It determines the target of all storage range, FORMAT, and dump ID operands until another address space operand or the end of the command string.

### PRImary

is the default and causes the information to be taken from the primary address space.

PRIMARY can be used only by XC virtual machines.

### SPACE *userid:space\_name*

causes the information to be taken from the address space identified by *userid:space\_name*, where *userid*: is the user ID of the owner of the address space followed by a delimiting colon (it is optional and defaults to the user issuing the command), and *space\_name* is the name of the address space. The space name is 1 to 24 characters long.

The CP QUERY SPACES command returns *userid:space\_name* in the "SPACE IDENTIFICATION" field.

SPACE can be used only by XC virtual machines.

### ALET *alet*

causes the information to be taken from the address space identified by the ALET, *alet*. The ALET is 1 to 8 hexadecimal digits long. Leading zeros are optional.

ALET can be used only by XC virtual machines.

### ASIT *asit*

causes the information to be taken from the address space identified by the ASIT, *asit*. The ASIT is 1 to 16 hexadecimal digits long. Leading zeros are optional.

The CP QUERY SPACES command returns *asit* in the "ASIT" field.

ASIT can be used only by XC virtual machines.

### AREG *areg*

causes the information to be taken from the address space identified by the ALET in the access register *areg*. The register can be specified in hexadecimal characters (leading zeros are optional) or in decimal digits.

AREG can be used only by XC virtual machines.

### *hexloc1*

0

is the starting guest real address to be dumped. The address *hexloc1* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability, leading zeros are optional. If the value you specify for *hexloc1* is not a multiple of 4 KB, the system rounds down the value to a 4 KB boundary. The value you specify must be within the specified address space, and not greater than the maximum address that VMDUMP can dump (see Usage Notes "[11](#)" on page 2078). If you omit the *hexloc1* operand, the default is zero.

See Usage Note "[12](#)" on page 2078 for examples of range specifications.

### *-hexloc2*

**-END**

**-ALL**

**:*hexloc2***

**:END**

**:ALL**

is the ending guest real address in the range of storage locations to be dumped. END is the default. The address *hexloc2* may be 1 to 16 hexadecimal digits, separated with an optional underscore for readability, leading zeros are optional. The value you specify for *hexloc2* must be equal to or greater than *hexloc1*, within the specified address space, and not greater than the maximum address that VMDUMP can dump (see Usage Note "[11](#)" on page 2078). If the value you specify for *hexloc2* is equal to the starting location, the system dumps the single page of storage containing that location. If the

value you specify for *hexloc2* is not equal to the starting location and is not a multiple of 4 KB, the system rounds up the value to the next 4 KB boundary.

Specify END to dump the remaining contents of configured storage of the specified address space up to the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078). If you specify ALL, VMDUMP dumps from *hexloc1* to the end of the virtual machine address space up to the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078). A virtual machine address space consists of configured guest storage as well as discontinuous saved segments in use by the virtual machine. For address spaces that a user does not own, both END and ALL dump the contents of the entire address space up to the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078), including DCSSs.

If END or ALL was specified or defaulted and storage exceeds the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078), pages over that address will not be dumped. An information message describing this situation will be displayed.

If you specify only the dash (-) or colon (:) range delimiter, and omit either *hexloc2* or END, END is the default. However, if you omit both the range delimiter and *hexloc2* or END, the system dumps the single page of storage starting at *hexloc1*. If you omit both *hexloc1* and *hexloc2* or END, *hexloc1* defaults to zero, *hexloc2* defaults to END, and VMDUMP dumps all of configured guest storage up to the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078). (Note that in this case specification of the range delimiter is irrelevant.)

See Usage Note “12” on page 2078 for examples of range specifications.

#### **.bytecount**

#### **.END**

#### **.ALL**

is a hexadecimal number indicating the number of bytes of storage to be dumped, starting with the first byte at *hexloc1*. The value you specify for *bytecount* must be at least 1 and can be from 1 to 16 hexadecimal digits; leading zeros are optional. The ending storage address resulting from *bytecount* must be within the specified address space and not greater than the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078). If the ending storage address is not a multiple of 4 KB, the system rounds up the value to the next 4 KB boundary.

Specify END to dump the remaining contents of configured storage of the specified address space up to the maximum address that VMDUMP can dump (see Usage Note 10). If you specify ALL, VMDUMP dumps from *hexloc1* to the end of the virtual machine address space up to the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078). A virtual machine address space consists of configured guest storage as well as discontinuous saved segments in use by the virtual machine. For address spaces that a user does not own, both END and ALL dump the contents of the entire address space up to the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078), including DCSSs.

If END or ALL was specified or defaulted and storage exceeds the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078), pages over that address will not be dumped. An information message describing this situation will be displayed.

If you specify only the period (.) range delimiter and omit either *bytecount* or END, END is the default. However, if you omit both the range delimiter and *bytecount* or END, the system dumps the single page of storage starting at *hexloc1*. If you omit both *hexloc1* and *bytecount* or END, *hexloc1* defaults to zero, *bytecount* defaults to END, and VMDUMP dumps configured guest storage up to the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078). (Note that in this case specification of the range delimiter is irrelevant.)

See Usage Note “12” on page 2078 for examples of range specifications.

#### **T0 \***

#### **userid**

transfers the dump to the virtual card reader of the specified user ID. If you enter an asterisk (\*) after the keyword T0, or do not specify the T0 at all, the system sends the dump to the virtual card reader of the virtual processor from which you issued the VMDUMP command.

**SYSTEM**

transfers the dump to the virtual card reader of the user ID specified on the SYSTEM\_USERIDS configuration file statement. (This is the user ID designated to receive any CP abend dumps.) For further details, see the description of the SYSTEM\_USERIDS Statement in *z/VM: CP Planning and Administration*.

**FORMAT *vmtype*****FORMAT FILE**

specifies the type of virtual machine being dumped. For *vmtype*, specify the 1- to 8-character name of the system that was running in the virtual machine (for example, CMS, RSCSNET, or PVM). If you omit the FORMAT operand, the default is FILE.

The dump viewing facility uses the FORMAT operand to identify the type of system or virtual machine dumped.

**DCSS****DSS**

specifies that CP take a dump of all discontinuous saved segments associated with the primary address space and not within user storage. If the user has DCSSs that are wholly within user storage, these are not added to the dump in response to the DCSS/DSS option. If a DCSS is partly within user storage and partly outside, only those segments outside user storage are dumped in response to this option.

If the DCSS or DSS keyword is specified with no *hexloc1*, no *hexloc2*, and no range delimiter, it is assumed that no ranges within configured guest storage are desired. In this case, the defaults are not used and only the user's discontinuous saved segments are dumped.

You can use this option only for your own host-primary address space.

**XA**

has no effect. XA is retained only for compatibility with previous releases. All dumps are in z/VM format.

**\**dumpid***

is a 1- to 100-character (including any imbedded blanks and special characters) identifier that is associated with the storage dumped for each address space for which the operand is specified. When specifying *dumpid*, you must precede it with an asterisk (\*). The system eliminates any blanks between the asterisk and the first nonblank character of the dump ID. If you specify a dump ID that is more than 100 characters in length *after* the blanks following the asterisk are eliminated, the system truncates it on the right to a length of 100 characters. If specified, the dump ID must be the last operand on the command line.

**CLOSE**

specifies that CP should close the VMDUMP system data file currently open for the issuing user, if one has been created by a previous request in continuous output mode.

**CANCEL**

specifies that CP should close and purge the VMDUMP system data file currently open for the issuing user, if one exists.

**CONt**

specifies that VMDUMP is to operate in continuous output mode, leaving the output file open after each request, until explicitly closed or canceled, or until a request is processed after continuous output mode has been reset.

**NOCont**

specifies that at the completion of subsequent VMDUMP requests, the output file is to be closed.

**Usage Notes**

Combining operands on the VMDUMP command

1. The CONT, NOCONT, CLOSE, and CANCEL operands are not valid in conjunction with any other VMDUMP operands.

2. When invoking VMDUMP to dump guest storage in continuous output mode, certain restrictions apply to the use of other command operands:
  - The SYSTEM and TO operands, which pertain to ownership of the file and its structure, can be specified only when the file is opened, that is, on the first request not counting any requests specifying CONT, NOCONT, CLOSE, or CANCEL. Subsequent requests that merely repeat the current setting of the operand are ignored, but requests that attempt to change the setting of the operand (whether it was set explicitly or by default) are rejected with an error message.
  - Operands used to designate which areas of guest storage to dump are valid on any of the series of requests in continuous output mode. These operands are:
    - PRIMARY
    - SPACE *userid:space\_name*
    - ALET *alet*
    - ASIT *asit*
    - AREG *areg*
    - address-range
    - DCSS
  - The FORMAT and *\*dumpid* operands can be specified on more than one request, but only the last value specified for a particular address space is recorded.
3. You can specify more than one address range on a single VMDUMP command, and they can be in any order.
4. With the exception of the address space operands and the dump ID, you can specify the operands in any order. The dump ID must be the last operand in the string of operands begun by an address space operand.

The contents of a virtual machine dump

5. VMDUMP processing always creates and records a symptom record when creating a dump.
6. When you dump any address space, you get pages that have been referred to and the storage keys for those pages. In addition, VMDUMP dumps this information for your virtual machine (not necessarily in this order):
  - The program status word (PSW)
  - General registers
  - Control registers
  - Access registers
  - Floating point registers
  - Floating point control register
  - The TOD clock
  - The TOD clock comparator
  - The TOD programmable register
  - The processor timer
  - The prefix register
  - The prefix page(s) for all virtual processors dumped
  - The ASIT and the space ID (*userid:space\_name*) for the primary address space.
  - The host access list associated with the primary address space.

**Note:** VMDUMP dumps processor information for each defined virtual processor, up to 64 processors.
7. When using the collaborative memory management assist, the VMDUMP command dumps pages in the stable block-usage state only.

8. Each page of virtual machine storage is dumped only once to a file. If a page is specified in separate calls to the same address space or on more than one request in continuous output mode, only the first request for the page is honored; later requests are suppressed. Likewise, if a DCSS or DSS associated with an address space is specified on more than one request in continuous output mode, only the first request results in the DCSS being written to the dump file.
9. Dumps for a V=V virtual machine do not contain pages that have never been referred to. When a VMDUMP request for a V=V virtual machine contains only ranges within pages that have not been referred to, only the virtual processor information that is automatically dumped is placed in the dump (see Usage Note “6” on page 2077). In continuous mode, this information is added to the dump on the first request that opens the dump file. Therefore, subsequent requests to dump ranges within pages that have not been referred to do not result in additions to the dump file.
10. VMDUMP supports the DEFINE STORAGE CONFIG command and will dump any specified storage ranges in discontinuous configured storage up to the maximum address that VMDUMP can dump (see Usage Note “11” on page 2078).
11. The maximum address that VMDUMP can dump is X'7FFFFFFF' for ESA/390 and ESA/XC virtual machines, and X'7FFFFFFFFF' (512 Gigabytes) for z/Architecture and z/XC virtual machines.
12. This table shows the effects of different combinations of parameters.

Parameters	What gets dumped
<i>hexloc</i> - <i>hexloc</i> . <i>hexloc</i>	Storage 0 through <i>hexloc</i> from address space
<i>hexloc</i> : <i>hexloc</i> - <i>hexloc</i> .	Storage <i>hexloc</i> through highest configured address of the address space but not more than the maximum address that VMDUMP can dump.
<i>hexloc1</i> <i>hexloc1</i> : <i>hexloc2</i> where <i>hexloc1</i> equals <i>hexloc2</i>	Single page of storage containing <i>hexloc1</i>
: . -	All of configured storage of the address space but not more than the maximum address that VMDUMP can dump.
Address space operand not indicating the primary space, and END or ALL	All of addressable storage of the address space but not more than the maximum address that VMDUMP can dump.
Primary space and END	Entire contents of the primary address space, up to maximum address that VMDUMP can dump, not including DCSSs.
Primary space and ALL	Entire contents of the primary address space, up to maximum address that VMDUMP can dump, including DCSSs.
Address space operand but no address ranges, DCSS, or DSS	All of addressable storage of the address space but not more than the maximum address that VMDUMP can dump.
No address space operands, address ranges, DCSS, or DSS	All of configured storage of the address space but not more than the maximum address that VMDUMP can dump.
DCSS or DSS but no address range	If DCSSs do not exist for the primary space, error message 013E is issued. If DCSSs do exist, the DCSSs associated with the primary address space are dumped.

Parameters	What gets dumped
DCSS or DSS, an address space operand and address ranges	If DCSSs do not exist, only the address ranges specified are dumped. If DCSSs do exist, the DCSSs associated with the primary address space are dumped, and the specified address ranges up to the maximum address that VMDUMP can dump.

#### Address space operands

13. An ALET operand of ALET 00000000 indicates the primary address space. Access register 0 is always taken as containing X'00000000'.
14. For non-XC virtual machines, address ranges are always prefixed. For XC virtual machines:
  - Addresses in the primary address space are prefixed (PRIMARY or ALET 00000000 is specified). Low-address protection may apply and fetch-protection checking may be performed.
  - Addresses in any other address space are never prefixed (ASIT, SPACE, and ALETs other than X'00000000'). Low-address protection never applies, and fetch-protection checking is always performed.

#### The FORMAT operand

15. In continuous output mode, a file type is assigned to the dump file when it is opened. This file type is assigned by the FORMAT operand of the first address space dumped. It is not changed until the file is closed. As a result, until the dump file is closed, QUERY READER shows the initial FORMAT value, not the value specified last. To see the file type that the dump file receives when it is closed, use the QUERY VMDUMP command.
16. For certain programs running in virtual machines, the Dump Viewing Facility provides a special dump-processing format. The Dump Viewing Facility identifies these programs by checking the file type of the reader spool file created by VMDUMP. The file type is an assigned name that you should specify for *vmtype* on the FORMAT operand when dumping a virtual machine in which one of these programs is running. The IBM-defined programs and their assigned names are:

Program	Assigned name
APPC/VM VTAM Services	AVS
Customer Information Control System (CICS®/VM)	CICSVM
Conversational Monitor System (CMS)	CMS
Group Control System for Extended Architecture (GCS/XA)	GCS
VM/Pass-Through Facility Program Product	PVM
Remote Spooling Communication Subsystem (RSCS) Networking Program Product	RSCSV2
RSCS Networking Program Product Version 2	RSCSV2
RSCS Networking Program Product Version 3	RSCSV3
RSCS Networking for z/VM, function level 530	RSCSV3
Shared File System	SFS
Transparent Services Access Facility	TSAF

There may be additional assigned names, peculiar to your installation, for programs that are a part of your system. Ask your system programmer for a complete list of FORMAT names.

#### Continuous mode

17. If a continuous mode dump file is open when the VMDUMP NOCONT command is issued, that file remains open until completion of the next VMDUMP request that causes a write to the file. For this reason, to ensure that a new file is created in response to your VMDUMP request, first enter the

VMDUMP NOCONT CLOSE command if a noncontinuous dump is to be created, or the VMDUMP CONT CLOSE command if a new continuous mode file is desired. This ensures that any files still open from previous requests are closed, and that continuous and noncontinuous mode is set as desired.

18. If a continuous mode dump file for a particular virtual machine is open while that machine is logging off, or if a dump file for that virtual machine was very recently closed before logging off, the virtual machine may be prevented for a short time from logging back on again. While the system completes work resulting from the close of the dump spool file, the user receives message:

```
HCP361E LOGOFF/FORCE pending for user ...
```

in response to logon attempts. Logon should be possible within a few minutes.

#### DCSSs

19. When VMDUMP CONT is issued, the DCSSs that are loaded at that time are the only DCSSs that appear in the dump until VMDUMP NOCONT is issued. For example, if VMDUMP CONT is issued, then DCSS T1 is loaded, and then VMDUMP DCSS is issued, DCSS T1 is not included in the dump.
20. When you specify an unaddressable range (an inline range that starts above guest storage and does not fall within a discontinuous saved segment), an informational message is issued, and that range is ignored.

#### Using a virtual machine dump

21. If you specify TO SYSTEM, VMDUMP sends the dump file to the user ID of SYSTEM. However, if you specify only SYSTEM, the system sends the dump file to the user ID designated to receive dumps at your installation.
22. You can view the dump interactively using the Dump Viewing Facility. For more information on viewing the dump produced by the [VMDUMP Command](#), see *z/VM: Dump Viewing Facility*.
23. After the system has written the dump to a reader spool file, you can use the DUMPLOAD utility to load it from the spool file into a CMS file.
24. In general, use the VMDUMP command to create dump files that can be used with the Dump Viewing Facility, and use the DUMP command to create dumps that you intend to read without the aid of the Dump Viewing Facility. For more information, see [Dumping, Displaying, Altering Virtual Machine Storage](#) in *z/VM: Virtual Machine Operation*.
25. If an external security manager is installed on your system, you may not be authorized to enter the VMDUMP command with the TO option. For additional information, contact your security administrator.
26. Long-running VMDUMP commands can be halted by the user by entering #CP CPHX. Another user can halt a user during VMDUMP processing by using the CPHX or FORCE command. A TERM BRKKEY (normally set to PA1 key) will not halt VMDUMP processing.
27. The VMDUMP command does not dump vector registers.

## Responses

Response 1:

```
Command complete
```

is issued when the command has finished successfully.

## Examples

These examples are for a virtual machine in XC mode. Virtual processor information includes registers, processor addresses, and access lists.

1. To dump processor information for up to 64 virtual processors and the contents of the current primary address space, enter

```
vmdump
```



2. You can specify several address spaces. For example,

```
vmdump pri asit 0BGE9A4000000007 2000 space joe:space1 1000-5000
```

creates a dump including processor information for up to 64 virtual processors, the entire contents of the primary address space, the page at address 2000 of the address space identified by the ASIT 0BGE9A4000000007, and the pages at addresses 1000 through 5000 of the address space JOE:SPACE1.

3. `vmdump pri dcsc format cms space user1:sfs 10000-20000 pri *cms abend`

creates a dump including processor information for up to 64 virtual processors, the DCSSs associated with the current primary address space, and the pages at addresses 10000-20000 in the address space USER1:SFS. The format associated with the primary address space is CMS and the dump ID is "\*CMS ABEND". The format for address space USER1:SFS defaults to FILE and there is no dump ID for that address space.

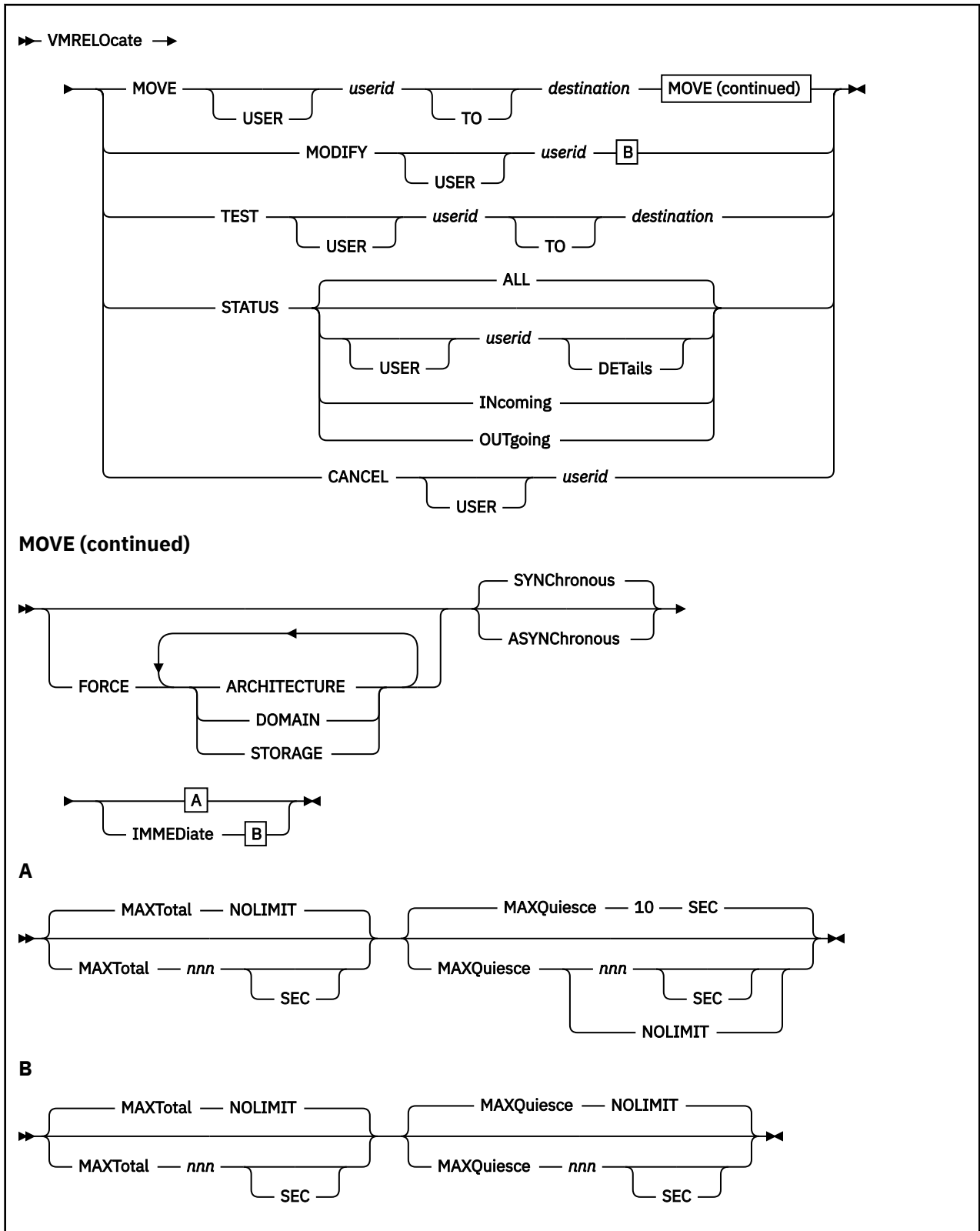
## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP004E Invalid hexloc - *hexloc*
- HCP005E Invalid hexdata - *hexdata*
- HCP007E Invalid userid - *userid*
- HCP009E Invalid range - *hexloc1-hexloc2*
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid
- HCP038E Parameter for *option* option missing or invalid - *parameter*
- HCP053E *userid/value* not in CP directory
- HCP146E VMDUMP failed; spooling error
- HCP160E Hexloc *hexloc* exceeds storage
- HCP161I END or ALL was specified or defaulted and storage exceeds maximum dumpable address (*hexloc*). Pages over that address will not be dumped.
- HCP162I Hexloc *hexloc* exceeds maximum dumpable address (*hexloc*) and will not be dumped.
- HCP356E Access denied; User *userid* file *spoolid* not {changed|transferred|printed}
- HCP410E CP ENTERED; PAGING ERROR
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP847E Maximum system spool file limit exceeded
- HCP1001E An operand is missing for ALET|AReg|ASIT|Space
- HCP1451I CP is unable to dump page *hexloc*
- HCP1452I The VMDUMP command is unable to process CPU dump data for more than 64 CPU's.
- HCP6150E An option has been specified that is not valid in {ESA/390|ESA/XC|z/Architecture|z/XC} mode.
- HCP6151E Storage {operand} is not addressable because of { a protection |an addressing | a segment translation | a page translation | a translation specification | an ASN translation | an AFX translation | an ASX translation | an ALET translation | an ALEN translation | an ALE sequence | an ASTE validity | an ASTE sequence } exception.
- HCP6156E {ASIT *asit*|Address space identifier *spaceid* |Access register *acc\_reg*|ALET *alet*} does not designate an address space to which you have the appropriate access.
- HCP6186I {Range *hexloc1-hexloc2*|DCSS or DSS} was previously dumped; it will not be dumped again.
- HCP6187E A soft abend occurred during VMDUMP processing.

## VMDUMP

- HCP6189I VMDUMP command is unable to add more data to the current virtual machine dump.
- HCP6190E VMDUMP processing is unable to complete due to a paging error.
- HCP6191I Storage range *hexloc1-hexloc2* is unaddressable and will not be dumped.
- HCP6525E Authorization request failed.
- HCP6577E Invalid ALET - *alet*.
- HCP6578E Invalid {access register|ASIT|address space identifier} - *option*
- HCP6790I VMDUMP processing halted.

# VMRELOCATE



**Note:** One or both of the MAXQUIESCE and MAXTOTAL options must be specified for a MODIFY command. In this case, there are no defaults for those options.

## Authorization

Privilege Class: A

## Purpose

Use VMRELOCATE to move an eligible, running z/VM virtual machine transparently from one z/VM system to another within an SSI cluster, and to monitor and cancel virtual machine relocations that are already in progress.

## Operands

### MOVE

moves the specified virtual machine, while it continues to run, to the specified system within the SSI cluster. The virtual machine being moved must be logged on to the system on which this command is issued.

### MODIFY

modifies the time limits associated with the relocation already in progress for the specified user. The virtual machine being moved must be logged on to the system on which this command is issued.

### TEST

tests the specified virtual machine and reports whether or not it is eligible to be relocated to the specified system. In the event that the specified virtual machine is not eligible for relocation, explanatory messages are displayed on the command issuer's terminal. The virtual machine being tested must be logged on to the system on which this command is issued.

### STATUS

displays information about relocations currently in progress. The STATUS operation can be initiated from either the source or the destination system.

### CANCEL

stops the relocation of the specified virtual machine. The virtual machine is not relocated to another system and will continue running normally, as though the relocation had not been attempted. The cancel operation can be initiated from either the source or the destination system. Once the request from the source system to start the virtual machine on the destination system has been sent, the relocation cannot be canceled.

The CPHX command can also be used to cancel a relocation. This is useful when VMRELOCATE was issued with the SYNCHRONOUS option. CPHX cannot be used to cancel an asynchronous relocation.

### USER *userid*

the userid of the target virtual machine for the VMRELOCATE command. Specify the USER keyword to make the following user ID unambiguous. The target userid cannot be the command issuer.

### TO *destination*

the SSI name of the destination system to which the specified virtual machine will be relocated or tested for relocation eligibility. Specify the TO keyword to make the following destination name unambiguous.

### ALL

specifies that the status of all relocations currently in progress on this system are to be displayed.

### INcoming

specifies that the status of only incoming relocations are to be displayed. An incoming relocation is a relocation resulting in a virtual machine being relocated from some other system to the system on which the STATUS command is issued.

### OUTgoing

specifies that the status of only outgoing relocations are to be displayed. An outgoing relocation is a relocation resulting in a virtual machine being relocated from the system on which the STATUS command is issued to some other system.

## Options

### FORCE ARCHITECTURE

indicates that relocation is to be attempted even though the virtual machine is currently running on a system with hardware architecture facilities, or CP-supplied features not available on the destination system, for example, when relocating to a system running an earlier release of CP. This option should only be used if it is known that the virtual machine and its applications do not use any facilities or features that are unavailable on the destination system. Other eligibility requirements, such as DCSS or device availability, will be respected and the relocation will be rejected if they are not met. Extreme care should be taken when using this option. If the unavailable hardware features or CP capabilities are important to the virtual machine or its applications, it is likely that the virtual machine or its applications will fail after the relocation is complete. This option is not needed to relocate a guest outside of its domain unless doing so would cause the guest's virtual architecture to lose facilities.

FORCE ARCHITECTURE is used in two circumstances:

- To allow out-of-domain relocation to a member that does not support all the current architectural features of the virtual machine.
- To allow a virtual machine to relocate to an excluded member of its domain.

### FORCE DOMAIN

indicates that relocation is to be attempted even though the virtual machine would be moved outside of its domain. This option does not permit relocation outside of the guest's domain if this move would cause the guest's virtual architecture to lose hardware facilities or CP capabilities. The FORCE ARCHITECTURE option must be used to make that kind of relocation occur.

**Note:** Both options (that is, FORCE ARCHITECTURE DOMAIN) must be specified to move a guest outside its domain if that move would cause a downgrade of the guest's virtual architecture.

### FORCE STORAGE

indicates the relocation should proceed if CP determines there are insufficient storage resources available on the destination for the following memory capacity assessment checks:

- Does the guest's maximum storage size (including CP supporting structures) exceed available space (central storage and auxiliary paging) on the destination?
- Does the guest's current storage size (including CP supporting structures) exceed auxiliary paging capacity on the destination system?
- Does the guest's maximum storage size (including CP supporting structures) exceed auxiliary paging capacity on the destination system?

The guest's maximum storage size includes any standby and reserved storage that could be configured for the virtual machine using dynamic storage reconfiguration.

These tests are performed several times during relocation. When FORCE STORAGE is not included on the VMRELOCATE command, then failure of any one of these tests results in termination of the relocation with appropriate message(s) (HCP1811I, HCP1812I, HCP1813I). Be sure to consider the size of the relocating guest along with the available storage resources and existing workload on the destination system prior to initiating the relocation. Use of the FORCE STORAGE specification with the VMRELOCATE command may increase your risk of a system outage.

Regardless of whether FORCE STORAGE is specified, relocation always checks to ensure the guest's current storage size (including CP supporting structures) fits in the available space (central storage and auxiliary paging) on the destination system. If relocation determines the guest won't fit, the relocation is terminated with message HCP1810I.

### ASYNChronous

causes the VMRELOCATE command to complete with a return code as soon as initial eligibility checks have passed and the relocation has started. This frees up the issuing virtual machine so that other commands can be entered while the actual relocation takes place.

### SYNChronous

causes the VMRELOCATE command to complete only after the relocation is complete. The success or failure of the operation is reflected in the return code. If SYNChronous is specified, the CPHX

command can be used to stop execution and effectively cancel the relocation. SYNCHRONOUS is the default.

**IMMEDIATE**

causes the VMRELOCATE command to do one early pass through virtual machine storage and then go directly to the quiesce stage. This option will likely decrease the total time required to move the virtual machine but, depending on guest activity, increase the amount of time the virtual machine is stopped during the quiesce stage. The defaults for both MAXTOTAL and MAXQUIESCE are NOLIMIT when IMMEDIATE is specified.

**MAXTotal nnnnnnnn SEC**

indicates the maximum total time (in seconds) that the command issuer is willing to wait for the entire relocation to complete. If the system determines that the relocation is taking longer than the specified total time then the relocation is canceled and the target virtual machine continues to run on the source system. Note that when VMRELOCATE MODIFY is issued without this option, the MAXTOTAL time is not changed to the default; instead the MAXTOTAL time remains unchanged from its previous value. The allowable values of nnnnnnnn are 1 to 99999999 seconds.

**MAXTotal NOLIMIT**

indicates that there is no limit on the total amount of time the system should allow for this relocation. The relocation is not canceled due to time limit constraints. This is the default.

**MAXQuiesce nnnnnnnn SEC**

indicates the maximum quiesce time for this relocation. This is the amount of time (in seconds) a virtual machine may be stopped during a relocation attempt. The relocation is canceled if the system determines the amount of time the virtual machine has been stopped is longer than *nnn* seconds. If the relocation is canceled for this reason, the virtual machine continues to run on the source system. The default for MAXQUIESCE is NOLIMIT if the IMMEDIATE option is specified or 10 seconds if not. Note that when VMRELOCATE MODIFY is issued without the MAXQUIESCE option, the maximum quiesce time is not changed to the default; instead the MAXQUIESCE time remains unchanged from its previous value. The allowable values of nnnnnnnn are 1 to 99999999 seconds. See usage notes for a more detailed explanation of this option.

**MAXQuiesce NOLIMIT**

indicates that there is no limit on the amount of time the system should allow for the quiesce phase of this relocation. The relocation is not canceled due to quiesce time limit constraints. This is the default if IMMEDIATE is specified. Otherwise the default is 10 seconds.

**Usage Notes**

1. A virtual machine cannot use the VMRELOCATE command to relocate itself.
2. The SET VMRELOCATE command can be used to indicate whether relocation is an option for this virtual machine and if so, in which relocation domain it is allowed to move. See the description of the SET VMRELOCATE command for more details. The QUERY VMRELOCATE command can be used to determine the current setting. The QUERY RELODOMAIN command can be used to determine which systems belong in the user's relocation domain, and therefore to which systems it is eligible to relocate.
3. Before a virtual machine is relocated, the capacity of the destination system should be carefully considered.
4. The number of concurrent relocations on one system should be limited as the overhead of a relocation can have an adverse effect on the performance of both the source and destination systems.
5. Relocation occurs in several steps. Much of the data transmission, in particular the virtual machine's storage (memory) contents, is done with the virtual machine still running so that the amount of down time can be minimized. Once most of the virtual machine storage has been transmitted, the virtual machine is stopped for the next step. Since there is likely to be a maximum period of time the guest can tolerate being unavailable, the MAXQuiesce option can be used to specify this period. If this is exceeded, relocation will be automatically canceled, and the virtual machine will resume operation on the original system.

6. Any CP monitor data that is being collected for this virtual machine on the source system will not automatically be collected on the destination system. If that data is required, the proper MONITOR commands must be issued on the destination system to start monitor for this virtual machine once it has been relocated.
7. If SET CPTRACE for this virtual machine is active, the relocation will occur but the CPTRACE setting will not carry over to the destination system.
8. APPLMON data collection will continue on the destination system without the need for manual intervention if diagnose x'DC' was used to start collection previously on the source system.
9. If the VM Resource Manager (VM/RM) is using Collaborative Memory Management (CMM) to adjust a Linux guest's storage utilization, the Linux virtual machine must be listed in the configuration file for VM/RM on both the source and destination systems.
10. Unpredictable results may occur if a relocation is done and there are local modifications to CP, in particular to the VMDBK.
11. When a virtual machine is relocated, its current share settings may or may not be appropriate on the destination system. Some thought must be given to what this user's share should be on the new system and the settings changed using the SET SHARE command if necessary.
12. A virtual machine's SET RESERVED setting is not preserved upon a successful relocation. The capacity and workload on the destination system should be considered when deciding whether to reestablish reserved frames for this virtual machine on the destination system.
13. If the virtual machine has any locked pages (via the LOCK command) they will be unlocked during relocation. Once relocation unlocks pages, they remain unlocked even if the relocation is subsequently terminated or canceled prior to completion.
14. Use the FORCE STORAGE option with caution. You should evaluate the amount of central storage, auxiliary paging capacity, and workload on the destination system to ensure you are confident the relocating guest will not exceed the capacity of the system. Also consult the z/VM CP Planning and Administration documentation for guidelines regarding paging space.
15. A guest that has been relocated should be given sufficient time to stabilize before being relocated again. Generally, this requires less than 60 seconds. More time may be required due to factors such as the number and type of devices being transferred, the number of outstanding I/O requests being transferred, and the CPU load and storage availability on the destination system. In particular, allow the guest to recover active OSA and FCP devices before initiating another relocation of the same guest.
16. If a relocation appears to be hung, use the VMRELOCATE STATUS command to determine if it is progressing. If not, the relocation may be delayed because the guest is in a state where intervention is required for the guest. An example of why the guest could be waiting is the password prompt on a CP LINK command.
17. When a guest on a virtual switch without priority queuing enabled relocates to a virtual switch with priority queuing enabled (and reversal), the guest's performance could be affected.
18. For relocation networking considerations, see [Live Guest Relocation Networking Considerations in z/VM: Connectivity](#).
19. APVIRT crypto users can relocate if there is an exact match in crypto capability between source and destination systems.

APVIRT crypto users can also relocate if there is not an exact match, and if the following conditions are true:

- APVIRT users are assigned to a relocation domain where the member systems have the Live Guest Relocation (LGR) for mixed APVIRT feature.
- The destination system is configured with a compatible shared crypto pool. Compatibility requires that crypto facilities that are presented to the guest at the source system are supported at the destination system.

APVIRT users that are assigned to a relocation domain where the member systems run LGR for mixed APVIRT see a virtual CEX adapter with the following characteristics:

## VMRELOCATE

- The virtual CEX adapter has characteristics that are common to all the members of that relocation domain that run shared pools of the same mode.
- The virtual CEX adapter presents the maximal common subset of capabilities for that mode.

For example, the subset of member systems of a relocation domain that run an accelerator mode shared pool present a virtual CEX adapter with accelerator capabilities. The adapter presents the maximal common subset of accelerator capabilities to any guest with a virtual accelerator. Member systems of a relocation domain that run a CCA-coprocessor mode shared pool present a virtual CEX adapter with CCA coprocessor capabilities. That adapter presents the maximal common subset of CCA coprocessor capabilities to any guest with a virtual CCA coprocessor.

For more information, see [Crypto Planning and Management](#) in *z/VM: CP Planning and Administration*.

20. The relocation process does not verify that the destination system has security keys for a secure IPL. If a guest is relocated to a system that does not have appropriate security keys, a secure IPL is not possible. Security keys are required on all members of a relocation domain where a guest might be relocated and later IPLed with the secure option. For more information, see [Guest Secure IPL](#) in *z/VM: Running Guest Operating Systems*.

## Responses

Responses issued by VMRELOCATE MOVE

This response is sent to the command issuer and system operator when different. The "by *userid*" text is only included in the system operator's version.

```
Relocation of user userid from source to destination
started (by userid) with FORCE ARCHITECTURE|DOMAIN|STORAGE in effect.
```

The following is sent to the command issuer and system operator when different. If the relocating guest generates a console file, this response is written to that file on both the source and destination systems.

```
User userid has been relocated from source to destination
```

```
Relocation of user userid from source to destination
did not complete. Guest has not been moved.
```

The following is sent to the command issuer and system operator when different.

```
Relocation of user userid from source to destination
did not complete. Guest has not been moved.
```

Response issued by VMRELOCATE TEST

```
User userid is eligible for relocation to destination
```

Response issued by VMRELOCATE CANCEL

This response is sent to user who canceled relocation, the user who started relocation if different, and the system operator if different.

```
Relocation of user userid from source to destination
has been canceled by userid on sysname_where_cancel_issued
```

Response issued by VMRELOCATE STATUS

```
No matching virtual machine relocation is currently
in progress on this system
```

User	From	To	By	Status	Elapsed
<i>userid</i>	<i>source</i>	<i>dest</i>	<i>userby</i>	<i>status</i>	<i>hh:mm:ss</i>



Response issued by VMRELOCATE STATUS *userid* DETails

User	From	To	By	Status	Elapsed
<i>userid</i>	<i>source</i>	<i>dest</i>	<i>userby</i>	<i>status</i>	<i>hh:mm:ss</i>

The following does not display when status = test.

```
Options: options
Max Total Time maxtotal
Max Quiesce Time maxquiesce

Total pages sent mmmmmmm in pp passes; sssss
pages sent in pass cc
```

Where:

***userid***

is the userid of the virtual machine being relocated.

***source***

is the system from which this virtual machine is moving.

***dest***

is the system to which this virtual machine is moving.

***options***

**synch, immed, or force.**

***userby***

is the user that initiated relocation for this virtual machine.

***status***

is the point in the relocation process that the virtual machine has currently reached. The value can be any of the following:

- Connecting - the source system is connecting to the destination system
- Elig Checks - relocation eligibility checking is in progress
- Creating Guest - Creating skeleton guest on destination system
- Moving Memory - The virtual machine's memory is being transferred to the destination system
- Stopping Guest - The virtual machine is being stopped on the source system
- Moving Guest - The virtual machine state is being moved to the destination system
- Final Mem Copy - Last pass of guest memory done to move the final changes to the destination
- Final I/O Check - Last I/O device eligibility checks done to ensure all necessary devices are still available on the destination
- Resume Guest - The virtual machine is being started on the destination system
- Cleaning Up - The relocation is finished and cleanup work is being done
- Terminating - The relocation had an error or was canceled and is in the process of terminating
- Test - a VMRELOCATE TEST is in progress for this virtual machine
- Comm Error - The VMRELOCATE STATUS command was issued on the destination system and a communications error occurred when attempting to retrieve the current status from the source system.

***hh:mm:ss***

shows a time in hours, minutes and seconds. This is the elapsed since the VMRELOCATE command was issued.

***synch***

indicates whether the VMRELOCATE command was issued synchronously or asynchronously. The value here will either be "ASYNCH" or "SYNCH".

**immed**

indicates whether the IMMED option was specified when the VMRELOCATE command was issued. *immed* will be displayed either as "IMMED YES" or as "IMMED NO".

**force**

indicates whether the FORCE option was specified when the VMRELOCATE command was issued. The FORCE option can be specified with one or more of several different options (ARCHITECTURE, DOMAIN, STORAGE). The FORCE output is displayed followed by some or all of those options, depending on which were specified on the command. For example, the Options line of output could look similar to any of the following:

```
Options: SYNCH IMMED YES FORCE ARCHITECTURE STORAGE
Options: SYNCH IMMED NO FORCE ARCHITECTURE DOMAIN
Options: SYNCH IMMED NO FORCE DOMAIN
```

**Max Total Time *maxtotal***

*maxtotal* can be either of the following two forms:

- **NO LIMIT** - which means the command issuer wants the relocation to complete with no limit on the total time it takes.
- ***n* SEC** - which is the maximum number of seconds the command issuer specified for the relocation to complete.

**Projected Total Time *n* Sec**

Projected total number of seconds to relocation completion. This is an estimate of the total time it will take to complete this relocation from start (command is issued) to finish (relocation completed). It is not the estimated remaining time. This may be zero if the VMRELOCATE STATUS command is issued early in the relocation and an estimate has not yet been calculated.

**Max Quiesce Time *maxquiesce***

*maxquiesce* can be either of the following two forms:

- **NO LIMIT** - which means the command issuer wants the relocation to complete with no limit on the amount of quiesce time it takes.
- ***n* SEC** - which is the maximum number of seconds the command issuer specified for maximum quiesce time for the relocation to complete.

**Projected Quiesce Time *n* Sec**

Projected number of seconds the virtual machine will be stopped during the processing of this relocation. This may be zero if the VMRELOCATE STATUS command is issued early in the relocation and an estimate has not yet been calculated.

**EXCEEDS TARGET**

shown after Projected Time or Projected Quiesce to indicate the projected time exceeds the user's specified target time.

***mmmmmm***

shows the total number of pages that have been transferred to the destination system thus far. (Note that an individual page may be sent more than once if it is changed between passes, and so an individual page may be counted more than once in this total.)

***pp* passes**

shows the total number of memory passes we have made for this relocation thus far. This number includes the current pass if one is in progress.

***ssss***

shows the total number of pages that were transferred in the memory transfer pass most recently completed.

***cc***

shows the number of the memory transfer pass most recently completed.

## Messages

- HCP003E *userid*: A failure occurred moving the content of the virtual machine's memory to the destination system
- HCP004E Relocation of *userid* to *destination* failed because a time out occurred while waiting for an expected response
- HCP013I *userid*: Using the maximum configurable size of the virtual machine would exceed available G auxiliary paging space on the destination system
- HCP026E Relocation is not in progress for *userid*
- HCP045E *userid* not logged in
- HCP263E Too many operands were supplied for this command
- HCP1161I *userid*: Virtual machine has a dedicated PCI function
- HCP1399E *userid*: Guest frame at virtual address *addr* was found still pinned during the last pass of memory move.
- HCP1802E *userid*: Message versions incompatible, message type = *type*, subsystem = *ssname*, source version = *sversion*, destination version = *dversion*
- HCP1803E *userid*: A failure occurred moving the content of the virtual machine's memory to the destination system
- HCP1804E Relocation of *userid* to *destination* failed because a time out occurred while waiting for an expected response
- HCP1805I *userid*: Guest is obtaining data from a service processor
- HCP1806E *userid*: The reserve status for DASD *vdev* could not be determined on the source system
- HCP1807E *userid*: The reserve held by DASD *vdev* cannot be relocated to the destination system
- HCP1807E *userid*: The reserve status for DASD *vdev* could not be determined on the destination system
- HCP1807E *userid*: The reserve held by DASD *vdev* cannot be relocated to the destination system because another system acquired it via an unconditional reserve
- HCP1808E Relocation of *userid* to *destination* failed because the destination is the same as the source
- HCP1810I *userid*: Current storage use *guest\_footprint* exceeds available capacity on destination *destination\_resource* by *difference*
- HCP1810I *userid*: Current storage use exceeds capacity on destination
- HCP1811I *userid*: Maximum storage use *guest\_footprint* exceeds available capacity on destination *destination\_resource* by *difference*
- HCP1812I *userid*: Current pageable storage use *guest\_footprint* exceeds available auxiliary paging space on destination *dest\_aux\_stg* by *difference*
- HCP1813I *userid*: Maximum pageable storage use *guest\_footprint* exceeds available auxiliary paging space on destination *dest\_aux\_stg* by *difference*
- HCP1820I *userid*: Virtual machine has directory authorization to use the virtual Multi-Path Lock Facility
- HCP1828I *userid*: A matching crypto facility does not exist on the destination
- HCP1924I *userid* Virtual machine is not in the destination system CP directory
- HCP1926E Relocation is not in progress for *userid*
- HCP1927E System *sysname* is not a member of an SSI cluster
- HCP1928E Cannot communicate with partner system *destination*
- HCP1929E Cannot relocate *userid* because *member* is not joined
- HCP1930I *userid*: The maximum *quiesce/total* time was exceeded
- HCP1931W Relocation of *userid* to *destination* can occur but the virtual machine may not be able to return to *source*

## VMRELOCATE

- HCP1932E Relocation of *userid* to *destination* is terminated as a result of a virtual machine action - *action*
- HCP1933E Cannot relocate *userid* because the SSI cluster is not STABLE or INFLUX
- HCP1934E A relocation MOVE|TEST of this virtual machine by *userid* is already in progress
- HCP1935E *userid*: Relocation processing error: RC=*code*
- HCP1936E *userid*: Device *vdev* is not responding
- HCP1937E *userid*: Unable to {ATTACH | DETACH | DEFINE | LINK | COUPLE} device *vdev*. RC=*code*
- HCP1938I *userid*: Virtual machine's storage configuration was created using the DEFINE STORAGE CONFIG command
- HCP1939E Relocation of *userid* to *destination* is terminated for the following reason(s): (This message is issued once, followed by one or more messages showing all the reasons why the specified relocation was terminated before completion.)
- HCP1940E *userid* is not relocatable for the following reason(s): (This message is issued once, followed by one or more messages showing all the reasons why the specified virtual machine was not eligible for relocation.)
- HCP1941E VMRELOCATE MODIFY is valid only on the system where the relocation originates
- HCP1942E *userid*: A virtual machine cannot issue a VMRELOCATE command with itself as the target
- HCP1943I *userid*: Virtual machine has SET VMRELOCATE OFF
- HCP1944I *userid*: Architecture incompatibility
- HCP1945I *userid*: Virtual machine is still logging on or is being autologged on
- HCP1946I *userid*: Virtual machine is logging off or is being forced off the system
- HCP1947I *userid*: Virtual machine is in system hold state on the destination system
- HCP1948I *use rid*: Virtual machine system reset is in progress
- HCP1949I *userid*: Virtual machine is primary system operator
- HCP1950I *userid*: Virtual machine is not running disconnected
- HCP1951I *userid*: Virtual machine has CP traces set
- HCP1952I *userid*: Virtual machine is terminating
- HCP1953I *userid*: Virtual machine has dedicated crypto capability
- HCP1954I *userid*: Virtual machine device *vdev* is a logical device
- HCP1955I *userid*: Virtual machine is IPLing
- HCP1956I *userid*: Single path CHPID virtualization is not enabled
- HCP1957I *userid*: Virtual machine is a batch worker machine
- HCP1958I *userid*: Virtual machine is a Coupling Facility virtual machine
- HCP1959I *userid*: Virtual machine configuration includes access to a virtual Coupling Facility
- HCP1960I *userid*: Virtual machine is using, or is the subject of, system trace facilities
- HCP1961I *userid*: Virtual machine is in a resource pool and a compatible pool does not exist on the destination system
- HCP1963I *userid*: Virtual machine is using VMCF
- HCP1964I *userid*: Virtual machine is using CMS POSIX facilities
- HCP1965I *userid*: Virtual machine has an Adjunct started
- HCP1966I *userid*: The integrated ASCII console is attached to the virtual machine
- HCP1968I *userid*: Virtual machine used DIAGNOSE x'214' to establish a pending page release and those page releases have not yet been processed
- HCP1969I *userid*: Virtual machine device *vdev* is a TDISK
- HCP1970I *userid*: Virtual machine device *vdev* is a public VDISK

- HCP1971I *userid*: Virtual machine has directory authorization to use diagnose x'98'
- HCP1972I *userid*: Virtual machine has a CPU-type or VCONFIG setting conflict
- HCP1973I *userid*: Virtual machine is in XC mode or has access to an address space owned by another virtual machine
- HCP1974I *userid*: Virtual machine has connections to another virtual machine or to a system service that cannot be relocated
- HCP1975I *userid*: Error loading NSS or DCSS *ssname* on the destination system; error code = *code*
- HCP1976I *userid*: Virtual machine is defined as multiconfiguration virtual machine
- HCP1977I *userid*: Virtual machine *vdev* has an open, non-console, SPOOL file
- HCP1978I *userid*: Virtual machine has an SPXTAPE operation in progress
- HCP1979I *userid*: Virtual machine is using DCSS that is a segment space or a member of a segment space - *ssname*
- HCP1980I *userid*: Identical NSS or DCSS *ssname* does not exist on the destination system
- HCP1980I *userid*: NSS or DCSS *ssname* on the source system does not have a checksum value
- HCP1980I *userid*: NSS or DCSS *ssname* on the destination system does not have a checksum value
- HCP1981I *userid*: Source system virtual device *vaddr* on real device *raddr* cannot be created on the destination system because the equivalent real device *raddr* is offline
- HCP1981I *userid*: Source system dedicated virtual device *vaddr* on real device *raddr* cannot be created on the destination system because the equivalent real device *raddr* is not free
- HCP1981I *userid*: Source system minidisk *vaddr* on real device *raddr* cannot be created on the destination system because the equivalent real device *raddr* is not system attached
- HCP1981I *userid*: Source system virtual device *vaddr* on real PAV device *raddr* cannot be created on the destination system because the equivalent real device *raddr* is not in the same PAV mode
- HCP1981I *userid*: Source system DEVNO defined minidisk *vaddr* on real device *raddr* cannot be created on the destination system because the equivalent real device *raddr* is neither a DEVNO defined minidisk nor free
- HCP1982I *userid*: Source system network device *vaddr* is coupled to a Guest LAN
- HCP1982I *userid*: Source system real network device *raddr* requires an EQID to be assigned
- HCP1982I *userid*: Source system network device *vaddr* is a simulated NIC coupled to a VSWITCH without external connectivity
- HCP1982I *userid*: Source system simulated network device *vaddr* is coupled to a VSWITCH without an assigned EQID
- HCP1982I *userid*: Source system simulated network device *vaddr* is coupled to a VSWITCH without external connectivity
- HCP1982I *userid*: Destination system VSWITCH *vswitch name* for simulated network device *vaddr* does not have external connectivity at this time
- HCP1982I *userid*: EQID of the destination system's VSWITCH *vswitch name* external OSA ports for the simulated network device *vaddr* does not match the source system's VSWITCH external OSA port EQID
- HCP1982I *userid*: Destination system VSWITCH *vswitch name* for simulated network device *vaddr* is not configured to have external connectivity or does not exist
- HCP1982I *userid*: Source system network device *vdev* is coupled to a VSWITCH but the virtual NIC has too many devices assigned
- HCP1982I *userid*: *nnn* consecutive free network devices with EQID *eqid* cannot be found on the destination system to relocate virtual device *vdev*
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured as IP for simulated NIC *vdevno*.
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured as ETHERNET for simulated NIC *vdevno*.

- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured as USERBASED for simulated NIC *vdevno*.
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured as PORTBASED for simulated NIC *vdevno*.
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured with NATIVE VLAN *vid* for simulated NIC *vdevno*.
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured with TYPE QDIO for simulated NIC *vdevno*.
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured with ISOLATION ON for simulated NIC *vdevno*.
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured with ISOLATION OFF for simulated NIC *vdevno*.
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured with VEPA ON for simulated NIC *vdevno*.
- HCP1982I *userid*: Destination system VSWITCH *swname* is not configured with VEPA OFF for simulated NIC *vdevno*.
- HCP1982I *userid*: Source system network device *vdevno* is coupled to a VSWITCH with VEPA ON. Destination system does not support VEPA.
- HCP1982I *userid*: Source system network device *vdevno* is coupled to a PORTBASED VSWITCH. Destination system does not accept PORTBASED relocations.
- HCP1982I *userid*: Source system network device *vdevno* is coupled with a PORTNUMBER outside the range supported by the destination system.
- HCP1982I *userid*: Source system network device *vdevno* is coupled to a VSWITCH with multiple network connections with different attributes. Destination system does not accept this configuration.
- HCP1982I *userid*: Destination system VSWITCH *vswitch name* for simulated network device *vaddr* does not support PRIQUEUEING. Change PQUPLINKTX setting to NORMAL.
- HCP1983I *userid*: Virtual machine's maximum storage size of *size* exceeds destination hardware maximum of *maxsize*.
- HCP1984I *userid*: Virtual machine's total VDISK space exceeds the VDISK system or user limit on the destination system.
- HCP1985I *userid*: The maximum number of users allowed is already logged on the destination system.
- HCP1986I *userid*: A list-directed IPL or dump is in progress
- HCP1989I *userid*: Virtual machine is hosting a logical device
- HCP1990I Relocation of *userid* has reset the virtual machine's SET RESERVED setting of *rrrrrrrr* to zero
- HCP1990I Relocation of *userid* will reset the virtual machine's SET RESERVED setting of *rrrrrrrr* to zero
- HCP1991I *userid*: *nnnnnnnn* guest pages locked to real storage frames have been unlocked during relocation
- HCP1991I *userid*: *nnnnnnnn* guest pages locked to host logical storage pages have been unlocked during relocation
- HCP1992I *userid*: A customizable field in a CP control block appears to be in use
- HCP1992I *userid*: Virtual machine device *vdev* has an associated CP control block that contains a customizable field in use
- HCP1993I *userid*: Virtual machine is relocating another virtual machine
- HCP1994I *userid*: Virtual machine has 370ACCOM set on
- HCP1995I *userid*: Virtual machine device *vdev* is an unsupported device
- HCP1995I *userid*: Virtual machine device *vdev* is an attached real terminal
- HCP1995I *userid*: Virtual machine device *vdev* is a dialed GRAF
- HCP1995I *userid*: Virtual machine device *vdevis* is a dialed in LINE

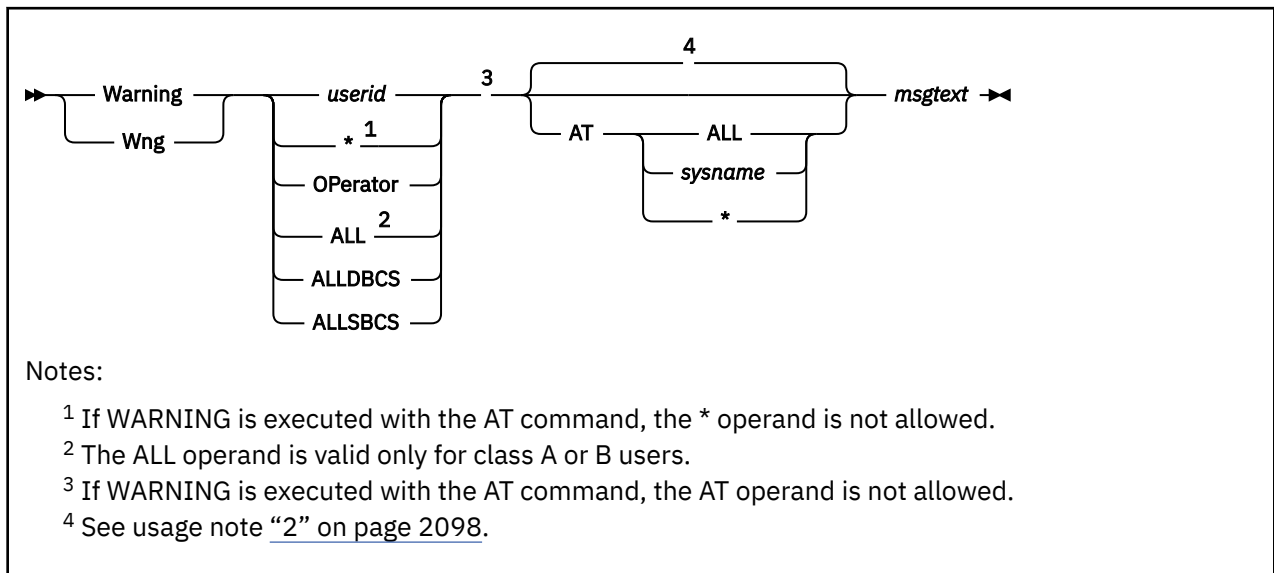
- HCP1995I *userid*: Virtual machine device *vdev* is a simulated CTCA that is coupled
- HCP1995I *userid*: Device *vdev* is coupled to a Coupling Facility virtual machine
- HCP1995I *userid*: Virtual machine device *vdev* is a dedicated SWITCH
- HCP1996I *userid*: Virtual machine device *vdev* is associated with a minidisk that was created with the DEFINE MDISK command
- HCP1996I *userid*: Virtual machine device *vdev* is a link to a private minidisk
- HCP1996I *userid*: Virtual machine device *vdev* is associated with a non-fullpack minidisk that was created with mode suffix V on the MDISK directory statement
- HCP1996I *userid*: Virtual machine device *vdev* is associated with a minidisk that is currently using virtual working allegiance simulation
- HCP1996I *userid*: Virtual machine device *vdev* is associated with a minidisk that is currently quiesced for HyperSwap
- HCP1996I *userid*: Virtual machine device *vdev* is an FBA DASD associated with real device *rdev* for which the reserve state is unknown
- HCP1996I *userid*: *userid*: Virtual machine device *vdev* is a full-pack overlay minidisk
- HCP1997I *userid*: Virtual machine device *vdev* is associated with real device *rdev* that has no EQID assigned to it
- HCP1997I *userid*: Source system virtual device *vaddr* on real device *raddr* cannot be created on the destination system because the real device does not exist there
- HCP1997I *userid*: Source system virtual device *vaddr* on real device *raddr* cannot be created on the destination system because an eligible equivalent real device cannot be found
- HCP1997I *userid*: Source system virtual device *vaddr* on real device *raddr* cannot be created on the destination system because its equivalent real device *rdev* is defined with a different device class and/or device type
- HCP1998I *userid*: Virtual machine device *vdev* has an outstanding delayed response message
- HCP1998I *userid*: Virtual machine device *vdev* is a single-user TAPE that is currently assigned
- HCP1998I *userid*: Virtual machine device *vdev* is a multi-user TAPE and the virtual machine currently owns the assign
- HCP1998I *userid*: Source system multi-user tape virtual device *vaddr* on real device *raddr* cannot be created on the destination system because the equivalent real device *raddr* is neither defined as a multi-user tape device nor free
- HCP1998I *userid*: Source system tape virtual device *vdev* on real device *rdev* is associated with an encryption key and cannot be created on the destination system because the equivalent real device *rdev* is associated with a different encryption key or one cannot be established for it
- HCP1998I *userid*: Virtual machine has directory authorization to use DIAGNOSE X'254' to control a Tape Library Dataserver
- HCP1998I *userid*: Source system tape virtual device *vdev* is a single-user TAPE for which assign status could not be determined
- HCP1998I *userid*: Virtual machine device *vdev* on real device *rdev* is not associated with an encryption key and cannot be created on the destination system because the equivalent real device *rdev* is associated with a key
- HCP2765E CP Exit *exit number* has returned unsupported return code *rc*
- HCP2779E CP Exit *exit number* has rejected the request
- HCP3150I *userid*: Destination system VSWITCH *swname* is reconfigured to GRANT *userid* for simulated NIC *vdevno*.
- HCP3150I *userid*: Destination system VSWITCH *swname* is reconfigured with PORTTYPE ACCESS for simulated NIC *vdevno*.
- HCP3150I *userid*: Destination system VSWITCH *swname* is reconfigured with PORTTYPE TRUNK for simulated NIC *vdevno*.

## VMRELOCATE

- HCP3150I *userid*: Destination system VSWITCH *swname* is reconfigured with PROMISCUOUS for simulated NIC *vdevno*.
- HCP3150I *userid*: Destination system VSWITCH *swname* is reconfigured with the source VLAN set for simulated NIC *vdevno*.



## WARNING



### Authorization

Privilege Class: A, B, C

### Purpose

Use WARNING to transmit a high-priority message to all or specified users.

#### Note:

1. WARNING does not wait for terminal input operations to complete. It breaks in on the line being displayed and sends the message immediately.
2. The SET WNG OFF command overrides this process.

### Operands

**userid**

\*

**OPERator**

**ALL**

identifies the user to whom you are sending the warning. To send a warning to yourself, specify an asterisk (\*).

Specify OPERATOR if you want to send a warning to the primary system operator, regardless of the operator's user ID. If you specify ALL, you broadcast the warning to all users receiving messages.

**ALLDBCS**

transmits the warning to all users logged on to DBCS (double-byte character set) terminals.

**ALLSBCS**

transmits the warning to all users logged on to SBCS (single-byte character set) terminals.

**AT ALL**

transmits the warning to all systems in the SSI cluster to which this system belongs. The lowest return code is returned to the command issuer.

**AT sysname**

transmits the warning to a specific system in the SSI cluster to which this system belongs. The return code is sent to the command issuer.

## WARNING

### AT \*

transmits the warning only to the system where the command is issued. This is the default if the system is not a member of an SSI cluster.

### *msgtext*

is the text of the warning message you are sending. The length of the message is limited by the number of characters remaining on the input line after the command and appropriate operand are entered. DBCS text is supported as message text.

## Usage Notes

1. If a warning is issued from a CMS environment, the command and the message text cannot be longer than 240 characters.  
If a warning is issued from a CP environment, the command and message length is limited by the input area of the terminal.
2. If the AT operand is not specified:
  - If WARNING is not executed with the AT command:
    - If the sender's system is not a member of an SSI cluster, the default is AT \*.
    - If the sender's system is a member of an SSI cluster, the default is AT ALL, except in the following cases, when the default is AT \*:
      - If the specified user ID is OPERATOR.
      - If the specified user ID is logged on to the same system as the sender.
      - If the specified user ID or the sender is defined by a multiconfiguration virtual machine definition.
      - If ALL, ALLDBCS, or ALLSBCS is specified, and a potential recipient user ID is defined by a multiconfiguration virtual machine definition, the default is AT \* for that user ID.
  - If WARNING is executed with the AT command, the AT operand function is bypassed (there is no default), and the message is sent to the system specified on the AT command.
3. The WARNING command routes the message to the secondary user of a disconnected virtual machine. If the secondary user is connected or the secondary user is disconnected but has a valid path to a \*MSG or \*MSGALL CP system service, the secondary user receives the message.  
If the secondary user is disconnected and there is no valid path to a \*MSG or \*MSGALL CP system service, the message *userid not receiving; disconnected* is returned to the command issuer.
4. If you receive a warning on a virtual console with the alarm feature, the alarm sounds.
5. If an external security manager (ESM) is installed on your system, you may not be authorized to enter this command. However, messages sent to or from the system operator and messages sent with the ALL option are not subject to authorization checking by the ESM. For additional information, contact your security administrator.

## Responses

Response 1:

```
hh:mm:ss * WNG FROM userid: msgtext
```

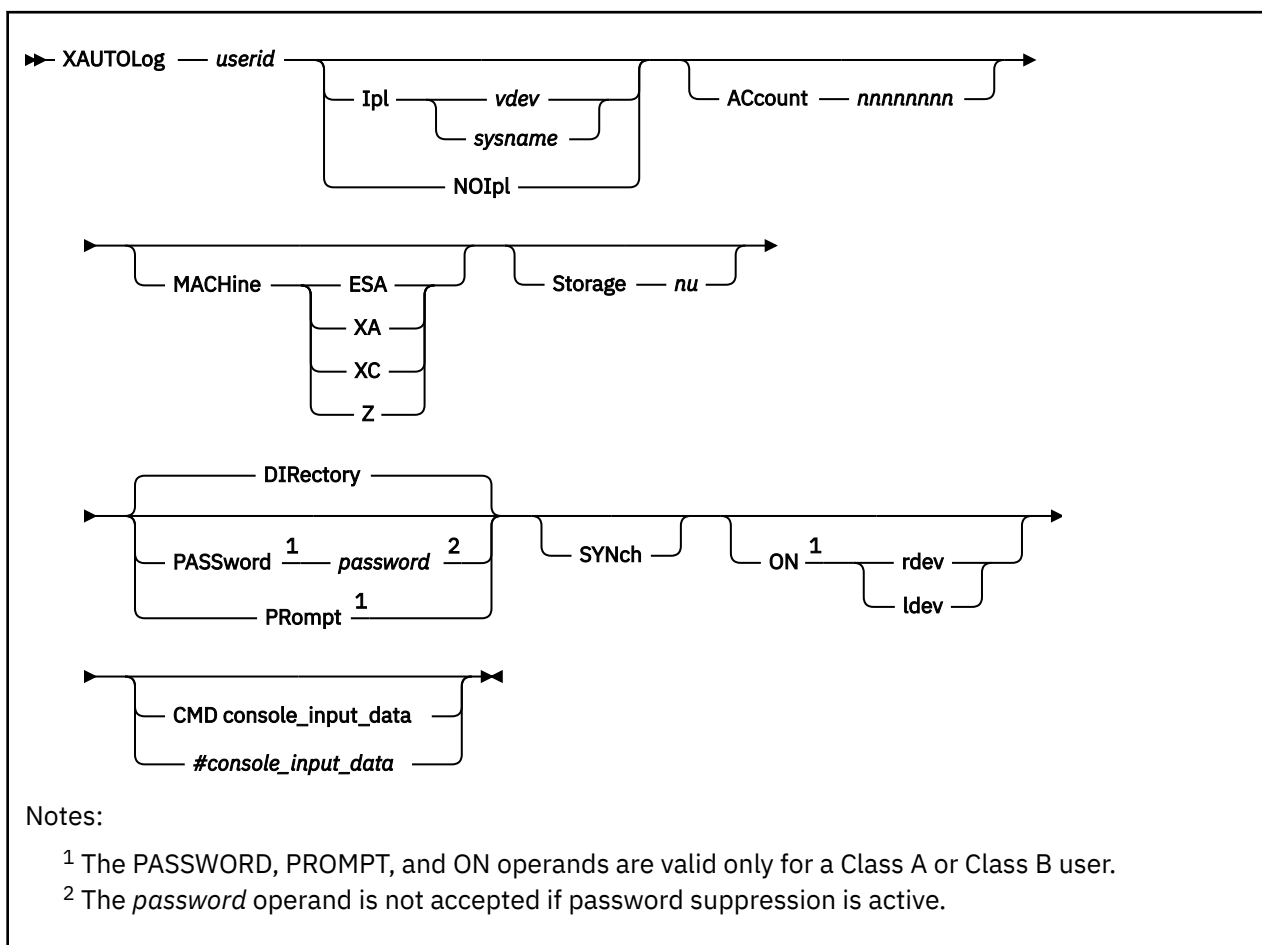
is the warning received by the specified users. The value *hh:mm:ss* represents the time the message was sent to the user.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP020E Userid missing or invalid
- HCP026E Operand missing or invalid

- HCP045E *userid* not logged on
- HCP057I *userid* not receiving; *text*
- HCP1142E The AT option is not allowed when this command is executed remotely with the AT command.
- HCP2970E The SSI cluster member name is missing or invalid.
- HCP2971I Specified SSI cluster member is not joined.
- HCP6600E An error was detected by installation-wide {*CP module name*|exit point 1210} - return code *nnnn*

# XAUTOLOG



## Authorization

Privilege Class: A, B, G

## Purpose

Use XAUTOLOG to log on another user virtual machine automatically.

**Note:** If you have Class G privileges only, the other user must have the appropriate XAUTOLOG statement in their system directory. If you have Class A or Class B privileges, and are not specifying a password, you are automatically authorized to log that virtual machine on.

## Operands

### *userid*

identifies the virtual machine you are logging on.

### **Ipl** *vdev*

### **Ipl** *sysname*

specifies the virtual device number or named saved system that you want IPLed after logon processing. This option overrides the IPL (if any) specified in the user's entry in the system directory.

If SYNCH is not specified, the IPL command processor determines whether the IPL operand is valid after the logon is completed; it is not verified as part of LOGON or XAUTOLOG processing.

If SYNCH is specified with IPL, the IPL operand is checked for validity before control returns to the XAUTOLOG command issuer.

The IPL command processor determines whether the IPL operand is valid after the logon is completed; it is not verified as part of logon processing.

### **NOIpl**

means that the IPL device specified in the user's entry in the system directory should not be used for an automatic IPL.

### **ACcount nnnnnnnn**

specifies an alternate account number to be assigned to the user's virtual machine at logon. This account number must be valid in order for the logon to complete successfully. If you do not specify this option, the primary (or first) account number specified in the user's entry in the system directory is used.

### **MACHine ESA**

### **MACHine XA**

### **MACHine XC**

### **MACHine Z**

specifies the mode (architecture) of the virtual machine. This option overrides the machine statement (if any) specified in the user's entry in the system directory. For further information on the [MACHINE](#) directory statement, see *z/VM: CP Planning and Administration*.

ESA designates ESA/390 (31-bit) architecture. A guest operating system can switch between ESA/390 and z/Architecture (64-bit) mode. XA is supported for compatibility and is functionally equivalent to ESA. XC designates ESA/XC architecture. A guest operating system can switch between ESA/XC and z/XC mode. Z designates a machine that executes only in z/Architecture mode.

See usage note “21” on page 2104.

### **Storage nu**

specifies an alternate storage size to be assigned to the virtual machine. The storage size is specified in the form *nu*, where *n* is a 1- to 8-digit decimal number and *u* is the 1-character storage unit suffix:

<i>Table 49. Maximum Input Values for Storage Units</i>	
<b>Storage unit suffix(u)</b>	<b>Maximum input value(n)</b>
K - kilobytes	99999999
M - megabytes	99999999
G - gigabytes	99999999
T - terabytes	16777216
P - petabytes	16384
E - exabytes	16

### **Notes:**

1. A K specification is rounded up to a MB value.
2. The maximum input value of 99999999 for the K, M, or G suffix is not a size limit but the physical limit of the operand (8 digits plus suffix). If the maximum input value for one of these suffixes does not allow you to define the amount of storage you want, you need to use a larger storage unit.
3. The value you specify for this option must be valid in order for the logon to be completed successfully. If you do not specify a value, the storage size defined in the user directory entry for the virtual machine is used. The maximum allowed value is the maximum storage size defined in the user directory entry for the virtual machine. The maximum cannot exceed a processor-specific limit imposed at run time.

**PASSword password**

is *userid*'s logon password. The ability to specify the password is available only to Class A or Class B users. Class A or Class B users are authorized to automatically log on any virtual machine without specifying a password, but this PASSWORD option is available when password verification is desired.

To determine whether password suppression is active for this command, use the QUERY PASSWORD XAUTOLOG command. If password suppression is active, you cannot specify a password on the command line. If you want to specify a password for the XAUTOLOG operation, use the PROMPT operand.

**PRompt**

specifies that you should be prompted for the password at the terminal. If password suppression is active for this command, you must specify PROMPT in order to use password authorization when entering the command at the terminal. The ability to specify PROMPT is available only to Class A or Class B users.

**DIRectory**

indicates that the user's entry in the system directory must contain an XAUTOLOG statement to authorize you to log on the virtual machine if you do not have Class A or Class B privileges.

**SYNch**

indicates that the command is to be run synchronously. CP returns control to the issuer of the XAUTOLOG command when the target user ID is logged on and IPL command is validated (if an automatic IPL is performed). Unless NOIPL is specified, there must be an IPL statement in the target user ID's directory entry or the IPL parameter must be specified on the XAUTOLOG command. (Note: If NOIPL is specified, CP returns control when the target user ID is logged on.)

If SYNCH is not specified, the XAUTOLOG command runs asynchronously, and CP returns control after a logon is scheduled. The XAUTOLOG command issuer receives a message when the target user is logged on. For asynchronous execution, no IPL statement or parameter is required.

**ON rdev**

specifies the real device number of a local 3270 device that is to be used as the console of the virtual machine. The ON option is available only to Class A or Class B users.

**ON ldev**

specifies the logical device number of the device that is to be used as the console of the virtual machine. Specify a logical device number as an "L" followed by up to four hexadecimal digits. Leading zeros may be omitted. The ON option is available only to Class A or Class B users.

**CMD console\_input\_data**

specifies the data that you want used as virtual console input in response to the first read issued by the user's virtual machine. You may use the remainder of the command line to enter console data.

**Note:** This option differs from *#console\_input\_data* in that logical line-end characters are not honored. Because of this, only one command can be sent. This option is intended to be used when the XAUTOLOG command is issued with the AT command to a remote member of the SSI cluster. It can also be used if the user being logged on is on the same system, with or without the AT command.

***#console\_input\_data***

is data that you want used as virtual console input in response to the first read issued by the user's virtual machine. You may use the remainder of the command line to enter console data; the entire line including any logical line-end characters (#) is passed to the virtual machine.

**Note:** # refers to the whatever logical line-end character is currently in effect for the virtual machine in which the XAUTOLOG command is being used.

If you are using CMS and you specify *#console\_input\_data* with the logical line-end character (#) on the command line, CMS truncates the XAUTOLOG command at the first logical line-end character. To prevent this, enter a logical escape character before the logical line-end character, or enter #CP, where # is your logical line-end character, before the XAUTOLOG command itself.

## Usage Notes

1. XAUTOLOG was a synonym for the AUTOLOG command. It was provided to ease future migration for those who wanted to use the extended options of the z/VM XAUTOLOG command.
2. A virtual machine is allowed to be logged on with XAUTOLOG even though the number of users already logged on is equal to or greater than the maximum allowed by the MAXUSERS operand of the FEATURES system configuration statement or by the SET MAXUSERS command if the user directory for the virtual machine was created using the IGNMAXU operand of the OPTION directory statement.
3. If neither IPL nor NOIPL is specified, the virtual machine that you log on using the XAUTOLOG command must have an IPL statement in its system directory entry unless it is a CF Service Machine (has OPTION CFVM in the directory).  
  
Unless NOIPL is specified, an automatic IPL is scheduled. If the IPL statement is not valid, you receive a message and *userid* is forced off the system.
4. Authorization to log on a user through the XAUTOLOG command is granted:
  - If you have Class A or Class B command privileges and you specify or default to the DIRECTORY option
  - If you have Class A or Class B command privileges and you specify a correct password in response to a prompt or through the PASSWORD option
  - If you have Class G (and not Class A or Class B) command privileges and your user ID appears in the XAUTOLOG directory statement belonging to the virtual machine that you want to XAUTOLOG.
5. Users with a password of NOLOG in their directory entries cannot be logged on by the XAUTOLOG command. For a complete description of the [XAUTOLOG directory statement](#), see *z/VM: CP Planning and Administration*.
6. If you do not specify the ON option, the virtual machine that you log on using the XAUTOLOG command operates in disconnect mode. The same restraints that apply to any disconnected machine also apply to virtual machines that are logged on with the XAUTOLOG command.
7. If you are using CMS and specify *#console\_input\_data* on the XAUTOLOG command line, CMS interprets the input line as two separate commands because of the logical line-end character (#). You can avoid this situation by turning off line editing, by preceding the logical line-end character with a logical escape character, or by preceding the XAUTOLOG command line with *#CP*, where *#* is your logical line-end character.
8. If the user ID you specify has a secondary user ID specified on the CONSOLE statement in its directory entry, messages from CP (and from the virtual machine if automatic IPL takes place) are routed to the secondary user if that user is logged on and not disconnected. If your user ID (XAUTOLOG issuer) is the designated secondary user ID and you are not running in disconnect mode, you receive messages resulting from the LOGON of the target user.
9. If your installation is using the password suppression facility for this command (controlled by the PASSWORDS\_ON\_CMDs operand of the FEATURES system configuration statement or by the SET PASSWORD command) and you want to use password authorization, you should enter the PROMPT parameter on the XAUTOLOG command line. Wait for the ENTER PASSWORD prompt before you type in the password. For system security, the password is automatically masked.
10. CP disables the XAUTOLOG and AUTOLOG commands for the rest of a user's log on session if a user enters excessive incorrect passwords to logon another user ID automatically. The initial system limit depends on how the JOURNALING system configuration statement is coded. Use the QUERY CMDLIMIT command to see the system limit value set by your installation. See *z/VM: CP Programming Services* for special considerations when issuing XAUTOLOG through the simulated virtual console interface (DIAGNOSE X'08').
11. If SYNCH is specified and an automatic IPL is scheduled but IPL processing is unsuccessful, you receive a message and the user ID is forced off the system.
12. If an external security manager (ESM) is installed on your system, you might not be authorized to use this command or you might not be authorized to use the ON operand for this command. For additional information, contact your security administrator.

13. There is no password authorization for an XAUTOLOG of a *userid* defined with the NOPASS operand, unless an ESM is installed. When an ESM is installed, password authorization may be required despite the NOPASS operand. Refer to documentation provided by your ESM for more information.
14. CP typically presents *console\_input\_data* to CMS through an interruption sequence before a CMS program has requested a console read. CMS places the *#console\_input\_data* into its terminal input queue in anticipation of some subsequent request to read from the console. This subsequent request is usually the PARSE EXTERNAL instruction in the SYSPROF EXEC, unless the SYSPROF EXEC is bypassed or the AUTOOCR parameter was used at IPL time. The EXTERNALS() VM REXX built-in function can be used to test for the existence of *console\_input\_data* before the PARSE EXTERNAL is issued.
15. Users defined with the LBYONLY operand in the password field of their USER or IDENTITY statement in the system directory may be restricted from performing functions that require password validation. When an ESM is not installed, users defined with the LBYONLY operand cannot be logged on by the XAUTOLOG command if password validation is required. When an ESM is installed, refer to the documentation provided by your ESM for more information.  
  
Refer to the [USER Directory Statement](#) or [IDENTITY Directory Statement](#) in *z/VM: CP Planning and Administration* for more details on the LBYONLY operand.
16. CP does not support System/370 architecture (370 mode) virtual machines. If a LOGON is attempted for a virtual machine whose directory entry specifies 370 mode, the XAUTOLOG command will be completed, but the virtual machine will be set to XA mode instead of 370 mode. Message HCP1512E is issued to inform the user.
17. If an XAUTOLOG is attempted for a virtual machine whose directory entry specifies z/Architecture-only mode, but the relocation domain does not support that mode, the XAUTOLOG command is completed, the virtual machine is set to ESA mode instead, and message HCP1512E is issued to inform the user.
18. If the ON option is specified, the designated real or logical device must exist and must be enabled.
19. If the ON option is specified and the virtual machine is already logged on, it will be reconnected on the designated device if it is disconnected or it will be disconnected from its current console device and reconnected on the designated device.
20. If the AT command is used to issue XAUTOLOG on a remote member of an SSI cluster, the PROMPT option is not allowed. If it is specified in this case, message HCP1516E is issued and the command is rejected.
21. An ESA (or XA) virtual machine will be put into either full ESA/390 mode or ESA/390 compatibility mode, depending on the level of ESA/390 capability available in the machine where the virtual machine is logged on or in the virtual machine's relocation domain. ESA/390 compatibility mode allows a subset of ESA/390 functionality sufficient for CMS and GCS, but lacks advanced functions like dynamic address translation (DAT). For details on ESA/390 compatibility mode, see [z/Architecture Principles of Operation](#).

## Responses

Response 1:

```
ENTER PASSWORD:
```

is the response to the PROMPT parameter. Enter the password of the user ID you want to XAUTOLOG.

Response 2:

```
Command accepted
```

indicates that the command syntax is correct.

This response appears if SYNCH is not specified and the command line syntax is accepted. This response means that the logon has been scheduled and VM has returned control to you. More information about the logon and IPL arrives later.



If the SYNCH parameter is specified, the COMMAND ACCEPTED response does not appear. All processing concerning the automatic logon is completed before VM returns control to you.

If an error occurs in trying to autolog the specified user, you receive an error message.

Response 3:

```
AUTO LOGON ***      userid USERS = nnnnn
```

is the response XAUTOLOG sends to the issuer of the command when that person is not the system operator. This response is generated by successful command completion.

\*\*\*

indicates there is no assigned line address for the virtual console.

**userid**

identifies the virtual machine initiated by XAUTOLOG.

**USERS = nnnnn**

is the number of logged-on users.

Response 4:

```
hh:mm:ss AUTO LOGON ***      userid USERS = nnnnn BY {SYSTEM }
                                     {userid2}
```

is the response XAUTOLOG sends to the system operator when the command completes successfully.

\*\*\*

indicates there is no assigned line address for the virtual console.

**userid**

identifies the virtual machine initiated by XAUTOLOG.

**USERS = nnnnn**

is the number of logged-on users.

**SYSTEM**

indicates that the XAUTOLOG process has occurred as part of VM system initialization. Typically, your primary system operator's virtual machine is logged on as part of the system initialization process.

**userid2**

identifies the user issuing the XAUTOLOG command.

Response 5:

```
hh:mm:ss USER DSC LOGOFF AS userid USERS = nnnnn FORCED
```

is the response to the system operator upon failure of the IPL for the user specified on the XAUTOLOG command. The system operator receives this response only if SYNCH is specified and an automatic IPL was scheduled for the *userid*.

**DSC**

indicates that the user you autologged is operating in disconnect mode.

**userid**

identifies the virtual machine initiated by the XAUTOLOG command.

**USERS = nnnnn**

is the number of logged-on users.

Response 6:

```
USER DSC LOGOFF AS userid USERS = nnnnn FORCED
```

is the response to the system operator if a not valid IPL statement is processed for the user specified in the XAUTOLOG command. The system operator receives this response only if an automatic IPL was scheduled for the user ID.

**userid**

identifies the virtual machine initiated by XAUTOLOG.

**USERS = nnnnn**

is the number of logged-on users.

Response 7:

```
AUTO LOGON type {rdev} userid USERS = nnnnn  
                {ldev}
```

is the response to the issuer of the command when that person is not the system operator and when the ON option of the XAUTOLOG command is specified.

**type**

is the display type.

**rdev****ldev**

is the real or logical device number where the user was logged on or reconnected.

**userid**

identifies the virtual machine initiated by XAUTOLOG.

**USERS = nnnnn**

is the number of logged-on users.

Response 8:

```
hh:mm:ss type {rdev} {AUTOLOGON} userid USERS = nnnnn BY userid2  
              {ldev} {RECONNECT}
```

is the response to the system operator when an XAUTOLOG command with the ON option specified completes successfully.

**type**

is the display type.

**rdev****ldev**

is the real or logical device number where the user was logged on or reconnected.

**userid**

identifies the virtual machine initiated by XAUTOLOG.

**USERS = nnnnn**

is the number of logged-on users.

**userid2**

identifies the user issuing the XAUTOLOG command.

## Messages

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP005E Invalid hexdata - <hexdata>
- HCP006E Invalid device type - <rdev | ldev>
- HCP013E Conflicting option - *option*
- HCP020E Userid missing or invalid
- HCP021E A real device number was not supplied or it is invalid
- HCP025E Storage missing or invalid
- HCP040E Device <rdev | ldev> does not exist
- HCP046E GRAF <rdev | ldev> is offline
- HCP049E GRAF <rdev | ldev> in use

- HCP051E GRAF <rdev | ldev> is not enabled
- HCP052E [XAUTOLOG failed for *userid*:] Error in CP directory
- HCP053E *value* not in CP directory [XAUTOLOG failed for *userid*:] *userid* not in CP directory
- HCP054E [XAUTOLOG failed for *userid*:] Already logged on {*line vdev*|*GRAF vdev* |*LUNAME vdev*|*device vdev*|*LDEV vdev*| disconnected|*SYSTEM sysname*|SYSC}
- HCP059E {XAUTOLOG|AUTOLOG} failed for *userid* - {IPL missing|IPL failed|accounting errors|incorrect password}
- HCP093E Storage size requested *reqsize* exceeds maximum allowed on this processor *maxsize*. Size set to maximum allowed.
- HCP115E {LINK|AUTOLOG|XAUTOLOG} invalid; excessive incorrect passwords
- HCP145I User *userid* at *rdev* issued an AUTOLOG command specifying *userid* with an invalid password *nnn* times. The limit is *limit*.
- HCP205I The virtual=real area is not available because it was not generated.
- HCP266E An account number was not supplied or is invalid.
- HCP361E [XAUTOLOG failed for *userid*:] LOGOFF/FORCE pending for user *userid*
- HCP475I Fatal I/O error trying to read directory from *valid* [for user *userid*]
- HCP780E [XAUTOLOG failed for *userid*:] Maximum password attempts exceeded, try again later.
- HCP1001E An operand is missing for *option*.
- HCP1100E A logical device number was not supplied or it is invalid
- HCP1472I One or more directory COMMAND statements were found for the CFVM user and were discarded. Directory COMMAND statements are not allowed for CFVM users.
- HCP1502E [XAUTOLOG failed for *userid*:] The directory entry for this *userid* is in use.
- HCP1509E You are not authorized to issue XAUTOLOG with the PASSWORD, PROMPT, or ON options.
- HCP1512E The System/370 machine mode was requested, but CP does not support System/370 mode virtual machines; the machine mode will be set to XA instead.
- HCP1512E The z/Architecture-only machine mode was requested, but is not supported in this relocation domain. The machine mode will be set to ESA instead.
- HCP1513E [XAUTOLOG failed for *userid*:] A logon is already in progress [on {*type rdev*|*type idev*|*liuname*|SYSC}]
- HCP1516E XAUTOLOG failed: The PROMPT option cannot be specified with the AT command.
- HCP6050E Your *userid* is not authorized to automatically logon *userid* *userid*
- HCP6052E XAUTOLOG command rejected; command format not valid.
- HCP6054E XAUTOLOG failed for *userid*: Explanatory message to follow.
- HCP6525E External Security Manager is unavailable.
- HCP6056I XAUTOLOG information for *userid*: The IPL command is verified by the IPL command processor.
- HCP6525E Authorization request failed.

## XLINK

---

### Authorization

Privilege Class: A, B

### Purpose

Use the XLINK family of commands to perform the following cross-system link functions:

#### XLINK CHECK

Use XLINK CHECK to determine whether one or more volumes are under cross-system link control.

#### XLINK FINDTAB

Use XLINK FINDTAB to attempt to verify that the CSE/SSI system table matches the date and time stamp in the CP table.

#### XLINK RESET

Use XLINK RESET to free links held by a failed system.

If a system is defined as a member of an SSI cluster, none of the XLINK commands are allowed.

### Usage Notes

1. The cross system extensions (CSE) environment is not supported. However, the cross-system link (XLINK) function that was included in CSE is still supported for non-SSI systems, and "CSE" is still used in some function and object names, command responses, and messages related to XLINK.
2. The following CP utilities provide additional XLINK functions:

#### XLINK DISPLAY

Use XLINK DISPLAY to display link indicators in the map records of all systems on a particular device. For more information, see [“XLINK DISPLAY” on page 2257](#).

#### XLINK FORMAT

Use XLINK FORMAT to prepare DASD for sharing by initializing the CSE area on the specified DASD volume. For more information, see [“XLINK FORMAT” on page 2260](#).

3. The following responses and messages can be issued for XLINK CHECK, XLINK FINDTAB, or XLINK RESET.

### Responses

```
HCP2883I  Volume volidn is not controlled by CSE LINK.
HCP2884I  Volume volidn is controlled by CSE LINK.
HCP2886D  Are you ready for XLINK RESET of
           system sysname, volume volidn?
           Enter YES or NO:
HCP2887I  XLINK RESET complete.
HCP2888I  XLINK RESET ignored.
HCP2892I  WARNING! XLINK RESET must be used with
           extreme caution. Improper use will cause
           loss of minidisk data.
```

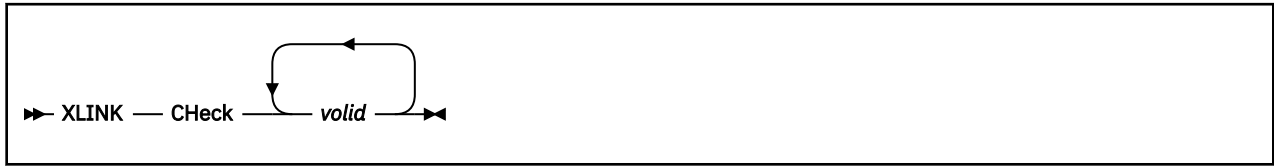
### Messages

- HCP0023E Volid missing or invalid
- HCP0026E Operand missing or invalid

- HCP1667E THE XLINK COMMAND IS NOT ALLOWED BECAUSE THE SYSTEM IS A MEMBER OF AN SSI CLUSTER.
- HCP2881E XLINK RESET is not allowed. There is no CSE LINK control on this system.
- HCP2882E XLINK RESET is not allowed. There is no CSE LINK control on system *sysname*.
- HCP2885E Volume *validn* is offline.
- HCP2889E XLINK RESET on your own system is not allowed.

## XLINK CHECK

---



### Authorization

Privilege Class: A, B

### Purpose

Use XLINK CHECK to determine whether one or more volumes are under cross-system link control. If a system is defined as a member of an SSI cluster, XLINK CHECK is not allowed.

### Operands

#### *valid*

is the 1- to 6-character name of a volume to be checked.

### Usage Notes

1. The cross system extensions (CSE) environment is not supported. However, the cross-system link (XLINK) function that was included in CSE is still supported for non-SSI systems, and "CSE" is still used in some function and object names, command responses, and messages related to XLINK.
2. The volume does not have to be mounted before you enter XLINK CHECK.

## XLINK FINDTAB

---

► XLINK — Findtab ◄

### Authorization

Privilege Class: A, B

### Purpose

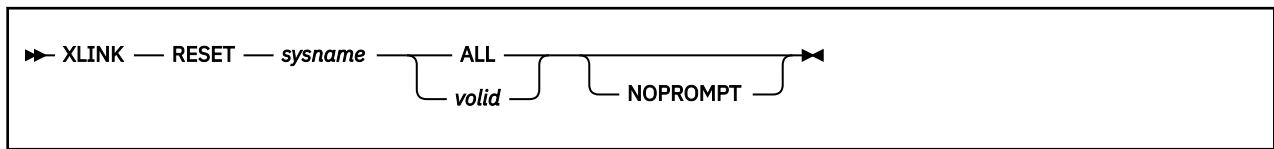
Use XLINK FINDTAB to attempt to verify that the CSE/SSI system table matches the date and time stamp in the CP table.

If a system is defined as a member of an SSI cluster, XLINK FINDTAB is not allowed.

### Usage Notes

1. The cross system extensions (CSE) environment is not supported. However, the cross-system link (XLINK) function that was included in CSE is still supported for non-SSI systems, and "CSE" is still used in some function and object names, command responses, and messages related to XLINK.
2. This function is performed during the XLINK FORMAT process. Under normal circumstances, you should not need to enter this command.

## XLINK RESET



### Authorization

Privilege Class: A, B

### Purpose

Use XLINK RESET only when you want to free cross-system links held by a failed system.



**Attention:** Issuing XLINK RESET when the specified system is not really down or is in the process of coming up destroys minidisk data—not directly, but by disrupting cross-system link operation.

If XLINK RESET *sysname* is issued and users have write access to one of the minidisks on a volume just reset, all link indicators are cleared. This condition can allow another user to get write access to the same minidisk, possibly causing a loss of minidisk data.

If a system is defined as a member of an SSI cluster, XLINK RESET is not allowed.

### Operands

#### *sysname*

identifies the name of the system whose map records are to be RESET. This system must *not* be in operation.

#### **ALL**

specifies that all volumes on XLINK\_VOLUME\_INCLUDE configuration statements for the system *sysname* are to be cleared.

#### *valid*

identifies the serial number or label of the volume to be cleared.

#### **NOPROMPT**

specifies that the RESET should be performed WITHOUT a system prompt to the user requesting that the user verify if the RESET should really be done.

### Usage Notes

1. The cross system extensions (CSE) environment is not supported. However, the cross-system link (XLINK) function that was included in CSE is still supported for non-SSI systems, and "CSE" is still used in some function and object names, command responses, and messages related to XLINK.
2. The system specified as *sysname* should not be considered down unless it is powered off or unless its storage has been cleared to binary zeros.
3. To avoid losing user data, enter the SYSTEM RESET and SYSTEM CLEAR commands for the system that is down. Do this before issuing the XLINK RESET command from another system.
4. This command clears the record of existing links held by the failed system, allowing users on the other systems identified on XLINK\_SYSTEM\_INCLUDE configuration statements to establish links to those minidisks.
5. The XLINK RESET command is not necessary for shutdown because the link indicators are automatically cleared.
6. To avoid problems, confirm that the suspect system is not operating before you enter this command.
7. You cannot enter the XLINK RESET command for the system you are logged onto.



---

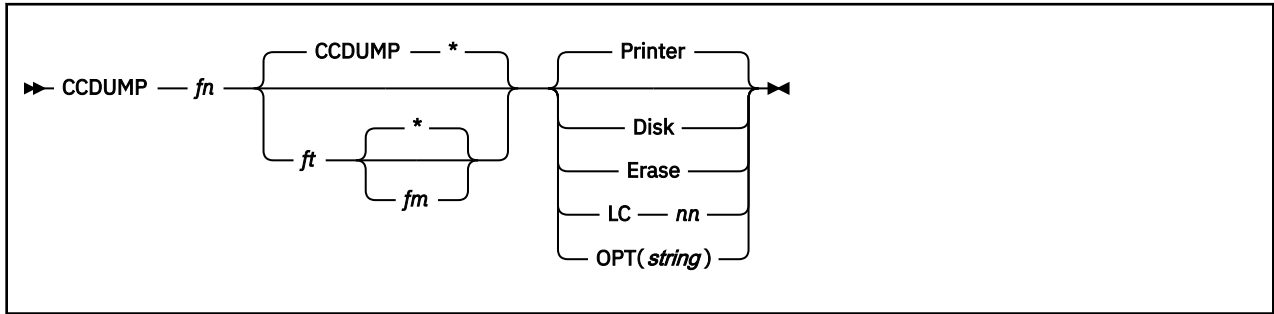
## Chapter 3. CP Utilities

This section contains reference information on the CP utilities. These utilities operate in the CMS environment. The utilities are listed in alphabetical order.

Each utility description is presented in the following format (some sections might not be included):

- **Name:** Identifies the name of the utility.
- **Format:** Shows the syntax of the utility with all the possible operands and options you can use.
- **Purpose:** States what the utility is used for.
- **Operands and Options:** Defines the function of each operand and option and any values you can include.
- **Usage Notes:** Identifies and describes special situations and other considerations that may affect your use of the utility.
- **Examples:** Provides one or more examples to show how the utility is commonly used.
- **Responses:** Describes the responses you might receive from the utility on your display device. Responses are normal operational output; they tell you about the execution and effect of the utility. Unlike system messages, utility responses are not prefixed with an identifying number and are not contained in *z/VM: CP Messages and Codes*.
- **Messages:** Lists the messages issued by the utility. Messages not unique to the utility might also be issued. Each message is prefixed with an identifying number. For more information on the messages, including suggested actions, see *z/VM: CP Messages and Codes*.

## CCDUMP



### Purpose

Use the CCDUMP utility to format and print a communications controller dump file produced by the CCLOAD command.

### Operands

#### *fn ft fm*

identifies the file containing a dump produced by the SSP utility IFLREAD, invoked directly or invoked by the CCLOAD command. The file type default is CCDUMP. The file mode default is asterisk (\*), indicating any disk. If a file name is not specified, CCDUMP processes the DUMP $nn$  file with the highest number  $nn$  on the specified disk, or all accessed disks if the filemode is specified with an asterisk (\*).

### Options

#### Printer

#### Disk

specifies the destination of the formatted dump output. Specifying PRINTER requires no A-disk space for the output file. Specifying DISK saves the formatted output on disk. The output file is always named *inputfn* LISTING A, where *inputfn* is the file name of the unformatted input file. The default is PRINTER.

#### Erase

specifies that the input dump file is to be erased after it is successfully printed.

#### LC $nn$

specifies the linecount parameter (the number of print lines per page) for the SSP utility IFLDUMP.

#### OPT *string*

specifies the optional formatting parameters to be passed to IFLDUMP in the SYSIN file. See the documentation for your communications controller for the allowable parameters. The string should not include the DUMP keyword, which is required and is generated automatically by CCDUMP. If a parameter string is not specified, the control statement contains just DUMP.

### Usage Notes

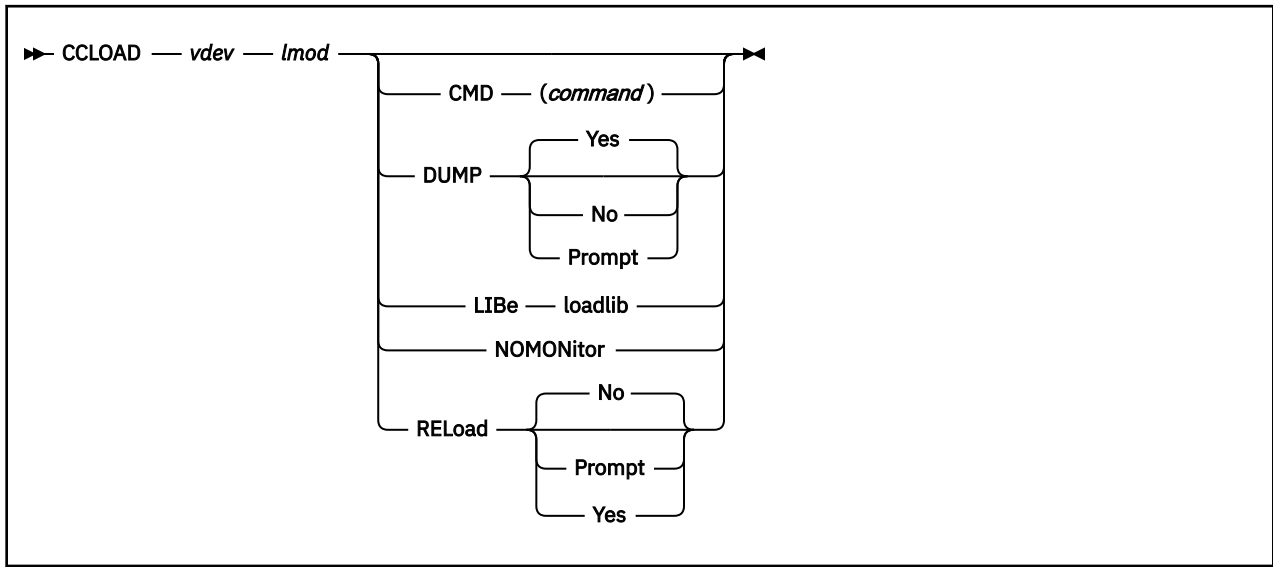
1. CCDUMP requires access to a R/W A-disk, even if PRINTER is specified.
2. If you specify DISK, you must have sufficient space on your A-disk for the formatted output file. Refer to the documentation for your communications controller for information on space requirements.
3. When you format the dump on the printer (PRINTER), messages produced by IFLDUMP do not appear on your console but are included in the SYSPRINT file. If you specify DISK, messages included in the SYSPRINT file are displayed on the console.

4. In order to erase the input dump file, you must have write access to the disk on which it resides. Do not specify the ERASE option if you are sharing access to files on another user's A-disk; for example, those generated by the CCLOAD EXEC on a disconnected service machine.

## Messages

- HCP8250E CCDUMP REQUIRES A R/W A-DISK
- HCP8255E THE COMMAND HAS AN INVALID USE OF "("
- HCP8256E *parm* IS AN UNRECOGNIZED KEYWORD OPERAND
- HCP8257E THE *kwd* KEYWORD OPERAND REQUIRES A VALUE
- HCP8258E *parm* IS AN INVALID VALUE FOR THE *kwd* KEYWORD OPERAND
- HCP8261I INVOKING *program*
- HCP8268E NO CTLR DUMP FILES WERE FOUND
- HCP8269E *program* FAILED WITH RETURN CODE *rc*
- HCP8270I CTLR DUMP FILE *fn ft fm* HAS NOT BEEN ERASED
- HCP8272I PROCESSING DUMP FILE *fn ft fm*
- HCP8276E CTLR DUMP *fn ft [fm]* WAS NOT FOUND

## CCLOAD



### Purpose

Use the CCLOAD utility to provide support for loading and dumping a communication controller manually, or from a service virtual machine.

### Operands

#### *vdev*

identifies the virtual device to which the base subchannel for loading and dumping the communication controller is dedicated or attached.

#### *lmod*

identifies the member of the emulation program (EP) load module to be loaded from the OS format LOADLIB created by the generation process.

### Options

#### **CMD *command***

specifies a command to be processed when the controller is loaded and ready for use. For example, you can notify the VM/Pass-Through Facility (PVM) to start its remote line drivers by specifying `cmd(cp smsg pvm exec ep01)` where EP01 PVM is a file containing PVM start commands.

#### **DUMP Yes**

#### **DUMP No**

#### **DUMP Prompt**

specifies whether the controller is automatically dumped when an EP control program failure is detected. If PROMPT is specified, the user is asked to decide whether to take the dump. If NOMONITOR is specified, the DUMP operand is ignored.

#### **LIBe *loadlib***

specifies the file name of the file type LOADLIB that contains the load module. If not specified, *lmod* is assumed.

#### **NOMONitor**

specifies that the command is to load the controller only, then end without monitoring controller and console interrupts. If not specified, the command monitors the controller and console for interrupts. A console attention interrupt results in prompting for a user decision on whether to dump and/or reload the controller, end the command, or ignore the attention.

**RELoad No****RELoad Yes****RELoad Prompt**

specifies the action taken when the controller is found to be already loaded. NO is the default and means that the controller is not reloaded. YES means that the controller is unconditionally reloaded. Prompt means that the user is prompted to decide whether to reload.

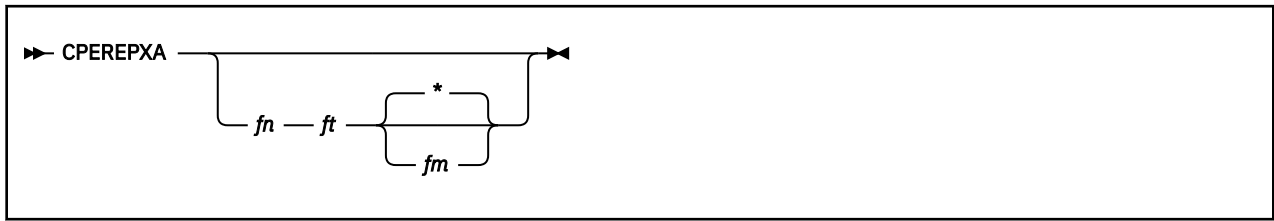
**Usage Notes**

1. CCLOAD requires access to a R/W A-disk. Loading requires space for SYSIN and SYSPRINT files. If CCLOAD is used for dumping the communication controller either automatically or at user request, it requires enough disk space for the dump files created by IFLREAD. For DASD requirements, refer to the Network Program Products publications.
2. Use the CCDUMP command to format and print the dump files produced by CCLOAD. If you provide another user with shared access to the A-disk of the user ID running CCLOAD, controller dump files can be printed off-line while the service virtual machine monitors the controller.
3. PROMPT should be specified on a disconnected machine only if you use the Single Console Image Facility to provide console support.
4. If NOMONITOR is specified, the DUMP operand is meaningless.

**Messages**

- HCP8250E CCLOAD REQUIRES A R/W A-DISK
- HCP8251E THE CTLR VIRTUAL DEVICE NUMBER *vdev* IS INVALID
- HCP8252E THE CTLR VIRTUAL DEVICE *vdev* IS UNDEFINED
- HCP8253E AN EP LOAD MODULE NAME IS REQUIRED
- HCP8254E *lmod* IS AN INVALID LOAD MODULE NAME
- HCP8255E THE COMMAND HAS AN INVALID USE OF "("
- HCP8256E *parm* IS AN UNRECOGNIZED KEYWORD OPERAND
- HCP8257E THE *kwd* KEYWORD OPERAND REQUIRES A VALUE
- HCP8258E *parm* IS AN INVALID VALUE FOR THE *kwd* KEYWORD OPERAND
- HCP8259E EP LOAD MODULE LIBRARY *fn* LOADLIB WAS NOT FOUND
- HCP8260E DEVICE *vdev* IS NOT A CTLR OR IS OFFLINE
- HCP8261I INVOKING *program*
- HCP8262I IGNORE *program* MESSAGE *msgno*
- HCP8263D MONITORING CTLR *vdev*. HIT ENTER TO STOP
- HCP8264D REPLY Y TO DUMP, OTHERWISE N
- HCP8264I CTLR *vdev* HAS FAILED
- HCP8265D MONITORING SUSPENDED FOR CTLR *vdev* REPLY [D TO DUMP,] L TO LOAD, Q TO QUIT, OR HIT ENTER TO RESUME
- HCP8266E UNABLE TO DUMP CTLR *vdev*. ALL DUMP FILES ARE IN USE
- HCP8267I CTLR *vdev* DUMPED TO FILE *fn* CCDUMP A1
- HCP8269E *progm* FAILED WITH RETURN CODE *rc*
- HCP8271D REPLY Y TO RELOAD, OTHERWISE N
- HCP8271I CTLR *vdev* IS ACTIVE
- HCP8273E A-DISK SPACE IS UNAVAILABLE FOR CTLR DUMPS
- HCP8274I A-DISK SPACE IS AVAILABLE FOR ONE CTLR DUMP
- HCP8275I EACH CTLR DUMP REQUIRES *nK*. AVAILABLE SPACE IS *nK*

## CPEREPXA



### Purpose

Use the CPEREPXA utility to access the Environmental Record Editing and Printing (EREP) program. This utility reads EREP records from a disk or a tape and produces printed output reports. The input records come from a CMS file that belongs to the error recording virtual machine or from a history tape. The content of the output report depends upon the specified or defaulted EREP keyword parameters and control statements and upon the EREP input records.

Because the EREP program is not used exclusively with z/VM, be sure that you have the latest publications that describe EREP before you attempt to invoke the program. See [Environmental Record Editing and Printing Program \(EREP\): User's Guide](#).

**Note:** The CPEREPXA utility produces error messages prefixed with the letters *HCP*. Messages issued from the EREP program are prefixed with the letters *IFC*. For descriptions of the HCP-prefixed messages, see [z/VM: CP Messages and Codes](#). For descriptions of the IFC-prefixed messages, see [Environmental Record Editing and Printing Program \(EREP\): Reference](#).

For information on [Setting Up Virtual Machines to Collect EREP Records](#), and an example of a procedure using the CPEREPXA utility to process EREP records, see [z/VM: CP Planning and Administration](#).

### Operands

#### *fn ft fm*

\*

is a control file containing EREP keyword parameters and control statements. (The keyword parameters and control statements are described in "EREP Keyword Parameters" below.) You can specify an asterisk (\*) in place of the file mode to indicate that the standard CMS search order is to be used to find the file.

If you do not enter the file name and file type of a control file, you are prompted to enter the keyword parameters and control statements from your display station.

#### EREP Keyword Parameters

You can enter the following list of EREP keyword parameters in your CPEREPXA control file or enter at your display station in response to the prompts issued by the CPEREPXA utility. This list contains only EREP keyword parameters. All control statements are passed to the EREP program without any recognition or action by CP.

This publication does not fully document the EREP program keyword parameters. It does not describe the precise meaning and use of the keyword parameter values that are required to be used with the keyword parameters. For complete details on EREP keyword parameters and control statements, including parameter-to-parameter relationships, refer to [Environmental Record Editing and Printing Program \(EREP\): User's Guide](#) and [Environmental Record Editing and Printing Program \(EREP\): Reference](#).

The EREP keyword parameters list contains two elements that are operands recognized by the CPEREPXA utility, CLEAR/CLEARF and TERMINAL. These operands are not operands for EREP, are unique to z/VM and are processed entirely by it. These operands are discussed only in this publication, and not in the [Environmental Record Editing and Printing \(EREP\) Program User's Guide and Reference](#).

*	MERGE
	MERGE=Y
	MERGE=N
ACC	
ACC=Y	
ACC=N	TERMINAL
	TERMINAL=Y
CLEAR	TERMINAL=N
CLEARF	
	THRESHOLD=(tempread,tempwrite)
ENDPARM	
	ZERO
HIST	ZERO=Y
HIST=Y	ZERO=N
HIST=N	

\*

in column 1 identifies a comment. Comments may appear anywhere in the input, regardless of whether it is entered from a control file or from your display station. EREP performs no processing; it simply places the contents of the comment in the TOURIST data set.

#### ACC=

indicates that selected error records are to be accumulated in an output data set. The particular error records selected and the source of these records (a CMS file, a history file, or both) depends on the other keyword parameters coded. The output accumulation data set is normally a tape mounted on tape drive 181, but this can be changed (see "CPEREPXA FILEDEFS" below). When output is accumulated on tape 181, the output is added as an extension of the existing file. The tape is rewound and spaced to the end of the first file before writing. Therefore, if you are using a tape for the first time, you should write a tape mark at the beginning of the tape before invoking CPEREPXA (the CMS TAPE command can do this). When output is accumulated on a tape, mount and ready the tape, and attach it to your virtual machine as tape 181 before invoking CPEREPXA. Note that for most types of reports, ACC=Y is the default.

**Note:** There are two CMS error record files: one for machine checks and one for other records.

#### CLEAR

#### CLEARF

This operand pair is exclusive to z/VM.

CLEAR requests z/VM to erase all error records from the error recording files, but not service record file (SRF) frame records. The CLEARF operand requests z/VM to erase all error records from the error recording files. Since z/VM does not record SRF frame records, these two operands are equivalent.

The CLEAR or CLEARF operand cannot be started along with keyword parameters; you must start each one in a stand-alone manner.

Because these operands cause error data to be erased from the input files, do not enter them within a sequence of keyword parameters or control statements that request error records as output. You should obtain all desired error data *before* issuing one of these operands.

#### ENDPARM

denotes the end of parameter specifications if you have used any keyword parameter statements. The ENDPARM keyword parameter is required if there are any control statements to enter, even if there are no keyword parameters to enter.

#### HIST=

indicates that the source of the error records for this run is to be a history data set rather than a CMS error record file. A history data set is a data set that was created as an accumulation (ACC) data set during an earlier session. Usually, the history data set is a tape mounted on tape drive 182, but this can be changed (see "CPEREPXA FILEDEFS" below). When input is from a history tape, mount and ready the tape and attach it to your virtual machine as tape 182 before invoking CPEREPXA.

**Note:** There are two CMS error record files: one for machine checks and one for other records.

**MERGE=**

indicates that the source of the error records for this run is to be both a history data set and a CMS error record file. The history data set is as described earlier for the HIST operand.

**Note:** There are two CMS error record files: one for machine checks and one for other records.

**TERMINAL=**

This operand is exclusive to z/VM.

When CPEREPXA encounters this operand while reading keyword parameters or control statements from a control file, CPEREPXA stops reading records from the file and begins to prompt for keyword parameters at your display station instead. All keyword parameters or control statements that are read from the control file before the TERMINAL operand is encountered are processed before CPEREPXA begins to prompt at your display station. Any subsequent records in the control file, however, are ignored. If CPEREPXA encounters this operand while reading from the display station, it is ignored. Control cannot be transferred back to the control file.

**THRESHOLD=**

Use the THRESHOLD keyword parameter to request a threshold report for the specified threshold values. The fields are not checked for syntax by CP. When CP recognizes the THRESHOLD keyword parameter, it changes the default value of the ACC keyword parameter to N.

**ZERO=**

Use the ZERO keyword parameter to tell EREP to clear the error recording files after generating the requested reports.

## Output Reports

For a complete description of the output reports that you can generate using CPEREPXA and the EREP program, see [Environmental Record Editing and Printing Program \(EREP\): User's Guide](#).

## Entering EREP Keyword Parameters and Control Statements

You must have IPLed CMS in your virtual machine in order to start the CPEREPXA utility. You can supply EREP keyword parameters or control statements to CPEREPXA by means of a console prompting technique, by a previously generated control file that contains the instructions required to produce the desired EREP record output, or by a combination of these two methods.

The sequence for invoking CPEREPXA is as follows:

1. Log onto the error recording service virtual machine. (Alternatively, log onto another virtual machine, link to the service virtual machine's minidisk that contains error records, and access that disk.)
2. IPL CMS.
3. Have the system operator attach any required tape devices to your virtual machine to serve for input and/or output data set use. (See the descriptions of the HIST and ACC operands).
4. Enter EREP keyword parameters or control statements through the file entry method or by the prompting method.

**Note:** The typical method of entering commands and operands on the same input line, as is done for other CP and CMS commands, is not valid. The reason that such action is disallowed is that many functions of EREP can exceed the maximum input line length the control program allows.

## Prompting Method

Type the CPEREPXA command at your display station and press the ENTER key. The system responds with this prompt: ENTER.

You may then enter EREP keyword parameters. If your needs exceed one line of input (limited by terminal line length), type a few keyword parameters and then press the ENTER key again. The system responds with the ENTER: prompt message again. You may then enter more keyword parameters. Repeat this process until no more keyword parameters are required. When this occurs, enter the ENDPARM keyword parameter to signal to the system that no more keyword parameters are to be entered.



Now you may enter control statements, but unlike the keyword parameters, they may be entered only one control statement for each line of input. Each time you press the ENTER key, the system responds with the ENTER: prompt message again. Repeat this process until no more control statements are required. When this occurs, press the ENTER key with no input. The null line signifies the end of the input.

When entering keyword parameters and control statements, observe the following rules:

- Separate keyword parameters and their associated values from the following keyword parameter by a blank (space), by multiple blanks, or by a comma.
- Enter embedded commas, periods, and parentheses that define the extent of variable keyword parameters as indicated in the CPEREPXA command format structure described earlier.
- Maintain the following sequence when entering input:
  1. Enter keyword parameters (if any).
  2. Enter the ENDPARM keyword parameter.
  3. Enter control statements (if any).

A comment card (\*) may appear anywhere in the sequence above. The TERMINAL operand is ignored if you enter it while using the prompting method.

- When you specify a keyword whose allowed values are Y and N, and you select the value Y, you may enter the keyword parameter without specifying a value. This form of the keyword parameter is always interpreted as a Y specification, regardless of the normal default value.
- To begin CPEREPXA with system default values, respond to the first ENTER: prompt message with the ENDPARM keyword parameter. When the second ENTER: prompt message appears, respond to it by pressing the ENTER key without any input; the null line signals to the system that there is no more input.

#### File Entry Method

The CPEREPXA command is entered followed by the file name, file type, and file mode of a file that contains a "package" of EREP keyword parameters and control statements arranged in the format as described in "Prompting Method" above. The same rules regarding blanks, commas, and parentheses apply; in addition, card images are truncated at column 71.

In practice, an installation probably has multiple files containing various operand mixes to satisfy the installation CPEREPXA report needs. To create and generate the necessary CPEREPXA files for this method of entry, use the CMS XEDIT command. File generation using the [CMS XEDIT command](#) is described in *z/VM: XEDIT User's Guide*.

#### Mixed Method

The CPEREPXA command is entered followed by the file name, file type, and file mode of a file that contains EREP keyword parameters and control statements, one of which is the TERMINAL operand of CPEREPXA. This operand may appear anywhere in the file, and keyword parameters (if any) and control statements (if any) are read until it is encountered. At this point, no further input is read from the file. Prompting begins at the display station, where additional EREP keyword parameters and control statements may be entered. When the system responds to the TERMINAL operand by transferring control from the control file to the display station, it expects input to proceed from the beginning of the sequence as it is described in "Prompting Method" above.

#### CPEREPXA FILEDEFS

CPEREPXA issues the FILEDEFS listed below before invoking EREP. These allow CMS to simulate the corresponding EREP files.

Where a mode letter of *x* is shown, *x* represents the read/write disk that has the most free space when CPEREPXA is invoked. At the end of the run, the FILEDEFS listed below are cleared with the exception of:

- EREPPT
- ACCIN
- ACCDEV.

These FILEDEFS are left intact because you can sometimes supply them yourself.

For those FILEDEFs where NOCHANGE is an option, you can supply an overriding FILEDEF before invoking CPEREPXA. The NOCHANGE option in CPEREPXA's FILEDEF means that it cannot change your previous FILEDEF.

```
filedef ereppt printer ( nochange blksize 133
filedef sysin disk sysin erepwork x3
filedef serlog disk serlog erepwork ( block 4096
filedef tourist terminal ( blksize 133
filedef directwk disk directwk erepwork x4
filedef accdev tap1 ( nochange recfm vb blksize 12000
filedef accin tap2 ( nochange recfm vb blksize 12000
```

### EREPPT

is EREP's printer file to which the report output is sent. You can override this FILEDEF with your own FILEDEF, which you can enter before invoking CPEREPXA.

### SYSIN

is a workfile built by CPEREPXA and read by EREP. You need not be concerned with it because it contains very few records. It is placed on the read/write disk having the most available space. It is erased automatically at the end of the run because its file mode number is 3. In those runs where there is no data to go into SYSIN, CPEREPXA issues FILEDEF SYSIN DUMMY for it, rather than the FILEDEF shown.

### SERLOG

represents the SYS1.LOGREC data set. The SYS1.LOGREC does not exist for z/VM; EREP's I/O to the SYS1.LOGREC is trapped by HCPIFC. Nevertheless, the FILEDEF is required to satisfy the needs of the OPEN and CLOSE commands issued by EREP because these are not trapped. Although a FILEDEF is defined, no corresponding file ever exists on any disk.

### TOURIST

is the message data set directed to your display station. EREP writes messages and diagnostics to this file with printer control characters. These characters may appear as data on the display station. A line of unknown characters may be printed before the EREP INFORMATIONAL MESSAGES heading.

### DIRECTWK

is a workfile built and read by EREP. It is not always created; whether it is created depends upon the particular report and whether the input comes from a history tape. This file may be quite large because it contains all input error records. The file is placed on the read/write disk having the most available space and is erased at the end of the run.

### ACCDEV

is the accumulation file. Normally, it is a tape on tape drive 181. This file is used if ACC=Y is specified either explicitly or implicitly. If ACC=Y is specified, tape 181 is rewound and spaced forward over the existing file and then backspaced over the tape mark before any writing is done. In this way, the tape is positioned to write new records at the end of the accumulation file.

**For standard label tapes only:** You must enter your own FILEDEF for ACCDEV before invoking CPEREPXA, so that the header labels are read correctly as *filedef accdev tapel sl (recfm vb blksize 12000*

In addition, if you want to update the standard label tape file, you should include the DISP MOD option.

**Note:** By issuing your own FILEDEF for ACCDEV before invoking CPEREPXA, you can override CPEREPXA's FILEDEF. In this way, you can accumulate data to another tape drive or to a disk file. However, the positioning of tape 181 is independent of your FILEDEF. This causes the following problems:

- Regardless of your FILEDEF, CPEREPXA attempts to position tape 181 as long as there is a tape 181 attached to the virtual machine (provided that ACC=Y).
  - If tape 181 is attached and ready, it is positioned.
  - If it is attached but not ready, the operator is notified, and CPEREPXA waits for the user to make it ready. The solution is to detach tape 181 before running CPEREPXA.

- CPEREPXA does not automatically position the user-defined file before writing into it, as it does when the file is tape 181.
  - If you define the file to another tape drive, the solution is to enter appropriate CMS TAPE commands to position the tape before invoking CPEREPXA.
  - If you define the file to a disk, the solution is to specify the DISP MOD option in your FILEDEF in order to add records at the end of an existing disk file.

You must always specify both RECFM and BLKSIZE. The record format must be either V or VB.

### ACCIN

is the history file. Normally, it is a tape on tape drive 182. This file is used if either MERGE=Y or HIST=Y is specified explicitly or implicitly. (HIST=Y is implied when certain reports are requested.) If either MERGE=Y or HIST=Y is specified, tape 182 is rewound before any reading is done.

**For standard label tapes only:** You must enter your own FILEDEF for ACCIN before invoking CPEREPXA, so that the header labels are read correctly.

**Note:** By issuing your own FILEDEF for ACCIN before invoking CPEREPXA, you can override CPEREPXA's FILEDEF. In that way, you can read history data from another tape drive or from a disk file. However, the rewinding of tape 182 is independent of your FILEDEF and can cause the following problems:

- Regardless of the your FILEDEF, CPEREPXA attempts to rewind tape 182 as long as there is a tape 182 attached to the virtual machine (provided that MERGE=Y or HIST=Y).
  - If 182 is attached and ready, it is rewound.
  - If it is attached but not ready, the operator is notified and CPEREPXA waits for the user to make it ready. The solution is to detach tape 182 before running CPEREPXA.
- CPEREPXA does not automatically rewind the user-defined file (if it is another tape drive) before reading from it as it does for tape 182. The solution is to enter a CMS TAPE command to rewind the tape before invoking CPEREPXA.

You must specify both RECFM and BLKSIZE if the history file is on a tape. The record format must be either V or VB.

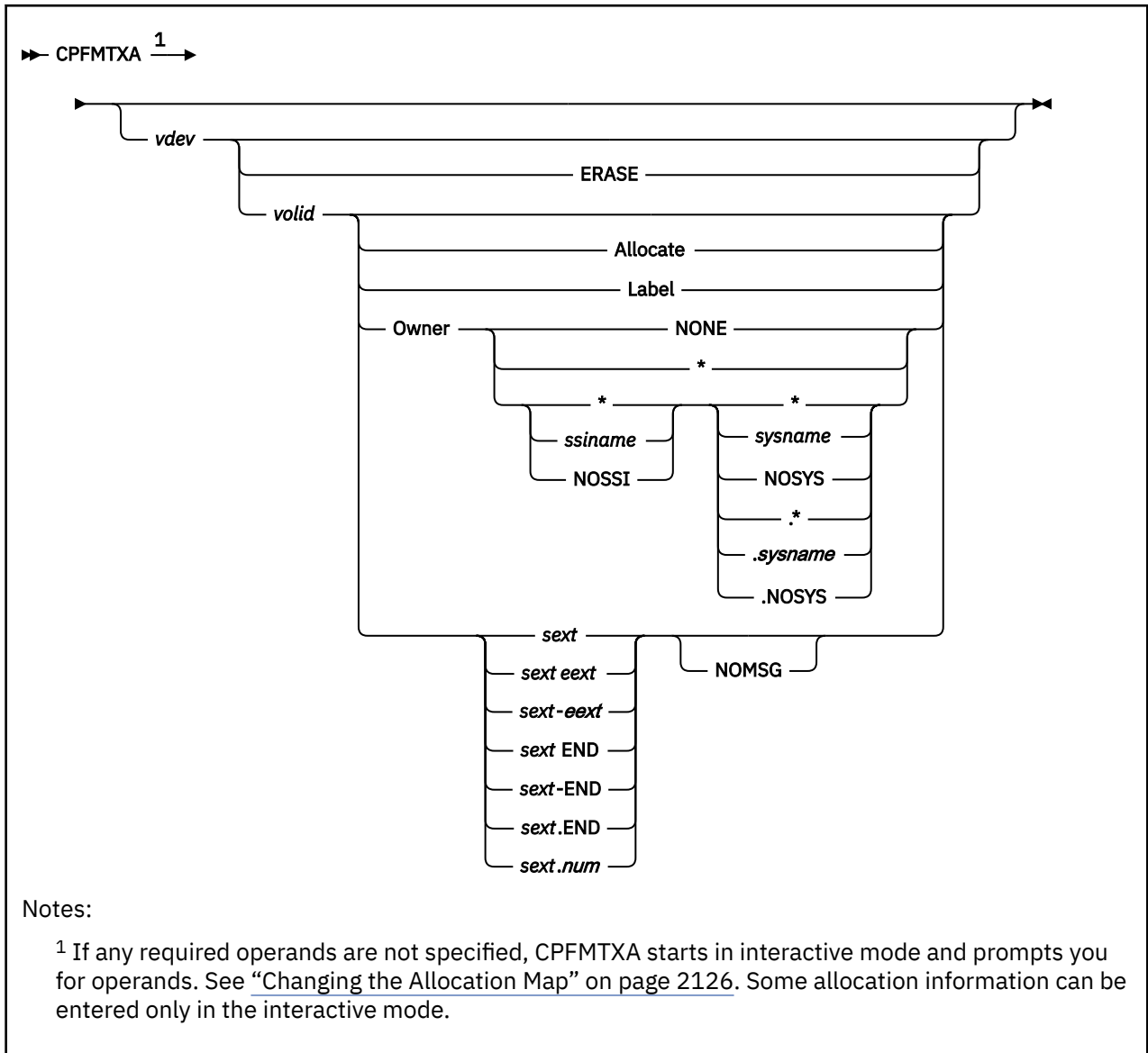
### Usage Notes

1. The CPEREPXA utility can invoke one or more CP commands that require a user privilege class other than G.

### Messages

- HCP863E Invalid parameter - *parameter*
- HCP2219E DISK *mode* NOT ACCESSED
- HCP8000E FILE *fn ft fm* NOT FOUND
- HCP8001E FILE *fn ft fm* IS NOT FIXED, 80 CHAR. RECORDS
- HCP8050E NO FILETYPE SPECIFIED
- HCP8051E MORE THAN 100 CHARS. OF OPTIONS SPECIFIED
- HCP8052E SOFTWARE INCOMPATIBILITY AT THE CPEREPXA-EREP INTERFACE. CODE=*code*
- HCP8053I CPEREPXA ZERO OR CLEAR HAS BEEN COMPLETED
- HCP8074E ERROR *code* READING FILE *file* FROM DISK
- HCP8075E CLEAR IS ONLY VALID WHEN SPECIFIED BY ITSELF
- HCP8077I ATTEMPTED ZERO WAS SUPPRESSED. REQUIRES R/W ACCESS

# CPFMTXA



**Notes:**

<sup>1</sup> If any required operands are not specified, CPFMTXA starts in interactive mode and prompts you for operands. See [“Changing the Allocation Map” on page 2126](#). Some allocation information can be entered only in the interactive mode.

## Purpose

Use the CPFMTXA utility to format, label, and allocate DASD volumes for CP uses such as paging, spooling, temporary disk, directory, and minidisk space. CPFMTXA can be used for both FBA and ECKD DASD. CPFMTXA requires ICKDSF, which is the standard method to format, label, and allocate DASD volumes for CP use. If ICKDSF is not available, no volumes are processed (see [Response 2](#)).

For information about using ICKDSF to format, label, and allocate DASD volumes, see [Device Support Facilities \(ICKDSF\): User's Guide and Reference](#).

## Operands

**vdev**

is the virtual device number of the DASD to be formatted, labeled, or allocated. It is specified as a 1- to 4-character hexadecimal number. The device type that is associated with this vdev must be supported as a CP-owned volume, and you must have write access to the device.

**ERASE**

indicates that you want to erase a volume. This option is not valid for FBA devices.

***valid***

is the volume identification label of the virtual device to be formatted, labeled, or allocated. *valid* is specified as a 1- to 6-character name. (The characters are not required to be alphanumeric.) Embedded blanks are not allowed. The volume ID is written on the disk if *sext* is 0 or if you specify the LABEL option. Otherwise, *valid* on the virtual device is read and compared to what you specify on the command line.

**Allocate**

indicates that you want only to reallocate a previously formatted volume.

**Label**

indicates that you want only to change the volume identification label. The volume must already be formatted. Only the volume identifier is changed; other information on the label record of the DASD remains unchanged.

**Owner**

indicates that you would like to specify or modify the ownership information for a CP-formatted device. Ownership information consists of an SSI cluster name and a system name.

**NONE**

indicates that the ownership information, if it exists, must be removed from the device.

**\***

***ssiname***

is the name of the SSI cluster that owns the device. Specifying an asterisk (\*) uses the name of the SSI cluster in which the current system is a member. If the current system is not a member of an SSI cluster, NOSSI is assumed.

The *ssiname* can be any 1- to 8-character alphanumeric string. However, OWNER *ssiname* \* and OWNER *ssiname*.\* are valid only when *ssiname* is the name of the SSI cluster in which the current system is a member.

Specifying OWNER \* is equivalent to specifying OWNER \*\* or OWNER \*.\*.

**NOSSI**

indicates that a null must be recorded for the SSI cluster name. OWNER NOSSI \* and OWNER NOSSI.\* are valid only when the current system is not a member of an SSI cluster.

**\***

**.\***

***sysname******.sysname***

is the system name of the system that owns the device. Specifying an asterisk (\*) uses the system name of the current system.

The *sysname* can be any 1- to 8-character alphanumeric string.

**NOSYS**

indicates that a null must be recorded for the system name.

***sext***

specifies the starting point of the extent to be formatted. The starting point is expressed as a cylinder number for ECKD DASD or a page number for a fixed-block device. The number is a 1- to 7-digit decimal integer (1- to 9-digit decimal integer for fixed-block devices) and must be within the size of the virtual device.

When *sext* is specified with no other range operand, only that cylinder or page is formatted.

*sext* can be specified with another range operand in any of the following formats:

***sext eext******sext-eext***

*eext* specifies the last cylinder (ECKD devices) or page (fixed-block devices) of the allocation. The *eext* cylinder number or page number has the following restrictions:

- *eext* is a 1- to 7-digit decimal integer (1- to 9-digit decimal integer for fixed-block devices).
- *eext* is equal to or greater than *sext*.
- *eext* must be within the size of the virtual device.

*sext* and *eext* can be separated by a dash or one or more blank spaces.

***sext* END**

***sext*-END**

***sext*.END**

END specifies that all remaining cylinders or pages of the volume are allocated.

*sext* and END can be separated by a dash, a period, or one or more blank spaces.

***sext.num***

*num* specifies a number of cylinders (ECKD devices) or pages (fixed-block devices) to be allocated. The allocation begins at *sext* and extends for *num* cylinders or pages. The *num* cylinder number or page number has the following restrictions:

- *num* is a 1- to 7-digit decimal integer (1- to 9-digit decimal integer for fixed-block devices).
- *num* is not 0.
- The sum of *sext* and *num* must not exceed the size of the virtual device.

*sext* and *num* must be separated by a period.

**NOMSG**

indicates that the ICKDSF cylinder progress messages are suppressed during formatting. Only the starting and ending cylinder message is displayed.

**Notes:**

- If you do not specify *sext*, LABEL, or ALLOCATE, then 0-END is assumed.
- If you do not specify any of the ALLOCATE, LABEL, OWNER, or *sext* options, the default is to format, allocate, and label the entire device.

**Changing the Allocation Map**

Whenever you format a disk or specify the ALLOCATE option, the current allocation of the disk is displayed as in the following example for an ECKD device:

```

CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:
TYPE      START      END      TOTAL
-----
type      nnnnn      nnnnn      nnnnn
:
type      nnnnn      nnnnn      nnnnn
    
```

For fixed-block devices, the current allocation is displayed as in the following example:

```

PAGE ALLOCATION CURRENTLY IS AS FOLLOWS:
TYPE      START      END      TOTAL
-----
type      nnnnnnnnnn  nnnnnnnnnn  nnnnnnnnnn
:
type      nnnnnnnnnn  nnnnnnnnnn  nnnnnnnnnn
    
```

The fields are described:

**type**

is normally DRCT, PAGE, PARM, PERM, SPOL, or TDSK.

**BAD**

indicates that the disk was not formatted by CPFMTXA. You can consider reformatting the entire disk by using CPFMTXA.

**nnnnnnnnnn**

is the cylinder or page range and total number of cylinders or pages that are allocated for each corresponding type.

You are then prompted to update the allocation map as in the following prompt for an ECKD device:

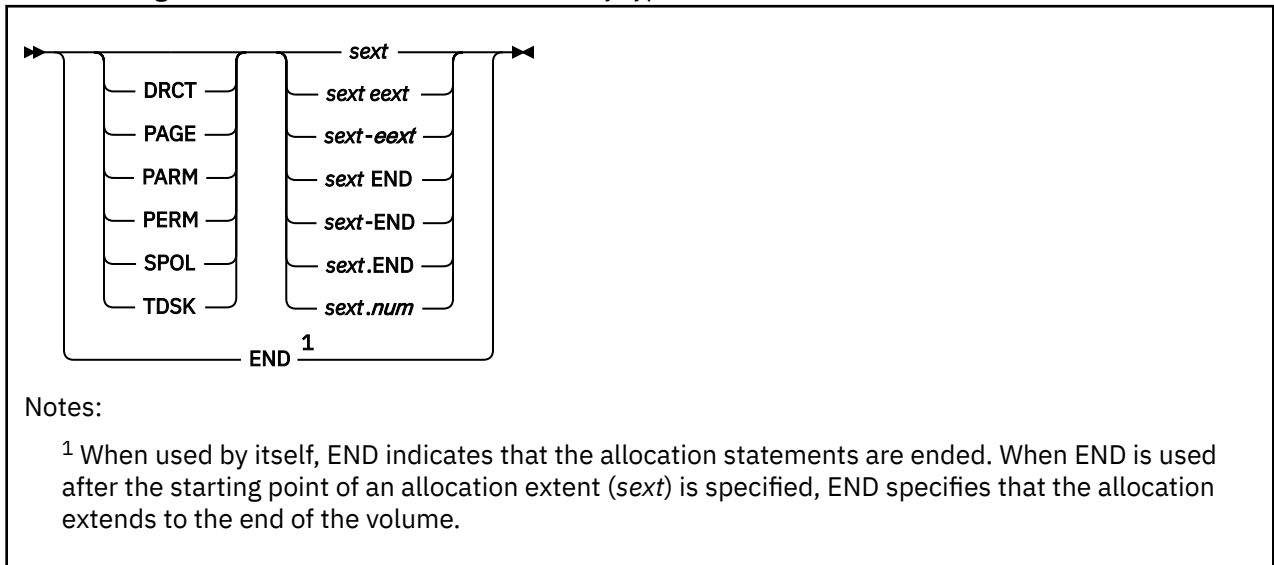
```
ENTER ALLOCATION DATA
TYPE CYLINDERS
.....
```

The following prompt is for a fixed-block device:

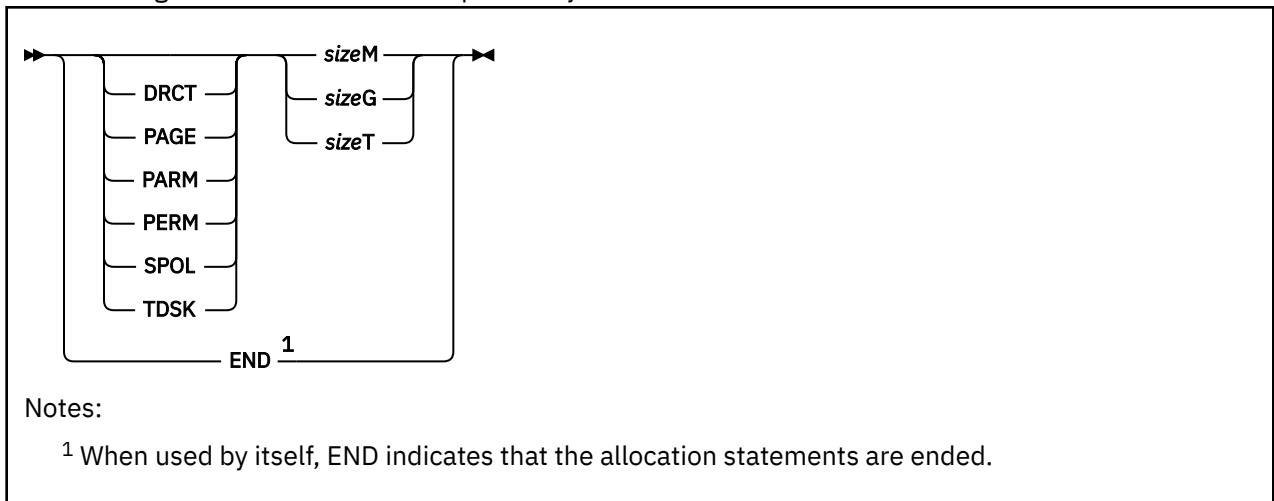
```
ENTER ALLOCATION DATA
TYPE PAGES
.....
```

Enter one line of data for each allocation.

The following allocation format can be used for any type of device:



The following allocation format is an option only for fixed-block devices:



The data elements are described:

**DRCT**

indicates allocation for directory files.

**PAGE**

indicates allocation solely for system paging activity.

**PARM**

indicates allocation that specifies the location of a CMS-formatted minidisk that contains the System Configuration files.

**PERM**

indicates allocation for CP user minidisks, checkpoint, warm start, and other static space.

**SPOL**

indicates allocation for spooling, dump space, or overflow paging activity.

**TDSK**

indicates allocation for temporary disk space, which becomes available to users after they log on to the system. Cylinder 0 cannot be allocated as TDSK.

**sext**

specifies the starting point of the allocation. The starting point is expressed as a cylinder number or page number. The number is a 1- to 7-digit decimal integer (1- to 9-digit decimal integer for fixed-block devices) and must be within the size of the real device where *vdev* is located.

When *sext* is specified with no other range operand, only that cylinder or page is allocated.

*sext* can be specified with another range operand in any of the following formats:

***sext eext******sext-eext***

*eext* specifies the last cylinder or page of the allocation. The *eext* cylinder number or page number has the following restrictions:

- *eext* is a 1- to 7-digit decimal integer (1- to 9-digit decimal integer for fixed-block devices).
- *eext* is equal to or greater than *sext*.
- *eext* must be within the size of the real device where *vdev* is located.

*sext* and *eext* can be separated by a dash or one or more blank spaces.

***sext END******sext-END******sext.END***

END specifies that all remaining cylinders or pages of the volume are allocated.

*sext* and END can be separated by a dash, a period, or one or more blank spaces.

***sext.num***

*num* specifies a number of cylinders or pages to be allocated. The allocation begins at *sext* and extends for *num* cylinders or pages. The *num* cylinder number or page number has the following restrictions:

- *num* is a 1- to 7-digit decimal integer (1- to 9-digit decimal integer for fixed-block devices).
- *num* is not 0.
- The sum of *sext* and *num* must not exceed the size of the virtual device.

*sext* and *num* must be separated by a period.

The following allocation examples are valid and yield the same allocation (2097120 pages of PERM) on a volume with an 8 GB (2,097,152 pages) capacity:

```
perm 32 2097151
perm 32-2097151
perm 32 END
perm 32-END
perm 32.END
perm 32.2097120
```



**sizeM****sizeG****sizeT**

is the size of the area to be allocated, in megabytes (M), gigabytes (G), or terabytes (T). The *size* operand is specified as a positive integer and must be immediately followed by M, G, or T.

An allocation that is expressed in this format is contiguous with the preceding allocation. If the first allocation is in this format, then 'PERM 0-31' is automatically included and the specified allocation starts at page 32.

Requested allocations are automatically reduced in some situations:

- The maximum extent for spool space is page 16,777,214 (64 GB). If you request a SPOL allocation that extends above page 16,777,214, then the allocation is automatically reduced in size so that the allocation extent ends at page 16,777,214 or the last page of the volume (whichever is less).
- If you request an allocation that extends beyond the volume capacity, then the allocation is automatically reduced in size so that the allocation extent ends at the last page of the volume.

**END**

When used by itself, END indicates that the allocation statements are ended.

**Interactions with ICKDSF**

If you enter CPFMTXA without operands and ICKDSF is available, you are prompted to enter formatting information. The following screens show examples of prompts, user input (in italic text) and responses.

**Example 1: Format and allocate an ECKD device by specifying the allocation ranges**

Allocations are requested by using the range format. This example is accurate for a count-key device (ECKD). For a fixed-block device, the prompts and responses would indicate allocation space in page units instead of cylinder units.

```

cpfmtxa
ENTER FORMAT, ALLOCATE, LABEL, OWNER, ERASE OR QUIT:

format

ENTER THE VDEV TO BE PROCESSED OR QUIT:

250

ENTER THE CYLINDER RANGE TO BE FORMATTED ON DISK 0250 OR QUIT:

0-544

ENTER THE VOLUME LABEL FOR DISK 0250:

vmpak1

CPFMTXA:
FORMAT WILL ERASE CYLINDERS 00000-00544 ON DISK 0250
DO YOU WANT TO CONTINUE? (YES|NO)

yes

HCPCCF6209I INVOKING ICKDSF.
ICK030E DEFINE INPUT  DEVICE: FN FT FM, "CONSOLE", OR "READER"
CONSOLE
ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"
CONSOLE
ICKDSF - CMS  DEVICE SUPPORT FACILITIES  14.0  (Con't)
      TIME: 09:33:50  01/31/91  PAGE 1

ENTER INPUT COMMAND:
  CPVOL FMT MODE(ESA) UNIT(0250) VOLID(VMPAK1) NOVFY RANGE(0,544)
  CPVOL FMT MODE(ESA) UNIT(0250) VOLID(VMPAK1) NOVFY RANGE(0,544)
ICK03020I CPVOL IS OPERATING IN VM/ESA MODE
ICK00700I DEVICE INFORMATION FOR 0250 IS CURRENTLY AS FOLLOWS:
  PHYSICAL DEVICE = 3390
  STORAGE CONTROLLER = 3990

```

```

STORAGE CONTROL DESCRIPTOR = CC
DEVICE DESCRIPTOR = 06
ICK04000I DEVICE IS IN SIMPLEX STATE
ICK03090I VOLUME SERIAL = SYSWK2
ICK03022I FORMATTING THE DEVICE WITHOUT FILLER RECORDS
ICK03011I CYLINDER RANGE TO BE FORMATTED IS 0 - 544
ICK003D REPLY U TO ALTER VOLUME 0250 CONTENTS, ELSE T
U
ICK03000I CPVOL REPORT FOR 0250 FOLLOWS:

    FORMATTING OF CYLINDER 0 STARTED AT: 09:33:52
    .
    .
    .
    FORMATTING OF CYLINDER 544 ENDED AT: 09:37:58

    VOLUME SERIAL NUMBER IS NOW = VMPAK1

    CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:
    TYPE      START      END      TOTAL
    ----      -
    PERM      0          2225    2226

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
    09:37:58    01/31/91

ENTER INPUT COMMAND:
END
END

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
ENTER ALLOCATION DATA
TYPE CYLINDERS
.....

perm 0.3
spol 3.300
tdsk 303 450
page 451-500
perm 501-end
end

HCPCCF6209I INVOKING ICKDSF.
ICK030E DEFINE INPUT DEVICE: FN FT FM, "CONSOLE", OR "READER"
CONSOLE
ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"
CONSOLE
ICKDSF - CMS DEVICE SUPPORT FACILITIES 14.0 (Con't)
    TIME: 09:40:27 01/31/91 PAGE 1

ENTER INPUT COMMAND:
    CPVOL ALLOC MODE(ESA) UNIT(0250) VFY(VMPAK1) -
    CPVOL ALLOC MODE(ESA) UNIT(0250) VFY(VMPAK1) -
ENTER INPUT COMMAND:
    TYPE((PERM,0,2) -
    TYPE((PERM,0,2) -
ENTER INPUT COMMAND:
    (SPOL,3,302) -
    (SPOL,3,302) -
ENTER INPUT COMMAND:
    (TDSK,303,450) -
    (TDSK,303,450) -
ENTER INPUT COMMAND:
    (PAGE,451,500) -
    (PAGE,451,500) -
ENTER INPUT COMMAND:
    (PERM,501,2225))
    (PERM,501,2225))
ICK03020I CPVOL IS OPERATING IN VM/ESA MODE
ICK00700I DEVICE INFORMATION FOR 0250 IS CURRENTLY AS FOLLOWS:
    PHYSICAL DEVICE = 3390
    STORAGE CONTROLLER = 3990
    STORAGE CONTROL DESCRIPTOR = CC
    DEVICE DESCRIPTOR = 06
ICK04000I DEVICE IS IN SIMPLEX STATE
ICK03090I VOLUME SERIAL = VMPAK1
ICK03024I DEVICE IS CURRENTLY FORMATTED WITHOUT FILLER RECORDS
ICK003D REPLY U TO ALTER VOLUME 0250 CONTENTS, ELSE T
U
ICK03000I CPVOL REPORT FOR 0250 FOLLOWS:

    CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:

```

```

      TYPE      START      END      TOTAL
      ----      -
      PERM      0          2          3
      SPOL      3          302         300
      TDSK      303         450         148
      PAGE      451         500          50
      PERM      501         2225        1725

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
          09:40:54      01/31/91

ENTER INPUT COMMAND:
END
  END

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0

```

**Example 2: Format and allocate an FBA device by specifying allocation sizes. Request less space than the volume capacity.**

The following example formats and allocates spool and page space on an FBA volume. The FBA volume can be a VDISK, EDEVICE, or real FBA device. 20 MB of SPOL space and 20 MB of PAGE space is requested on a volume with roughly 48.8 MB capacity. CPFMTXA allocates PERM space for the remainder of the volume.

The first step formats the volume. Because FBA devices use fixed-block architecture, their records do not need to be formatted. However, some formatting is required because the formatting process writes the label and default allocation map. Depending on how you plan to use the space, it might be prudent to use the format function to erase the disk contents. If you plan to use the disk for spool or page space, erasing the disk contents is unnecessary. The following example formats the first 256 pages of a larger 12,500-page FBA volume.

CPFMTXA generates some input for ICKDSF. In the example screens, user input is in italic text.

```

cpfmtxa
ENTER FORMAT, ALLOCATE, LABEL, OWNER, ERASE OR QUIT:

f
ENTER THE VDEV TO BE PROCESSED OR QUIT:

2234
ENTER THE RANGE TO BE FORMATTED ON DISK 2234 OR QUIT:

0-255
ENTER THE VOLUME LABEL FOR DISK 2234:

fb2234
CPFMTXA:
FORMAT WILL ERASE PAGES 000000000-000000255 ON DISK 2234
DO YOU WANT TO CONTINUE? (YES | NO)

yes
HPCPCF6209I INVOKING ICKDSF.
ICK030E DEFINE INPUT  DEVICE: FN FT FM, "CONSOLE", OR "READER"
CONSOLE
ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"
CONSOLE
ICKDSF - CMS/XA/ESA DEVICE SUPPORT FACILITIES 17.0
        TIME: 10:32:13          03/16/23          PAGE  1

ENTER INPUT COMMAND:
  CPVOL FMT MODE(ESA) UNIT(2234) VOLID(FB2234) NOVFY  -
ENTER INPUT COMMAND:
  RANGE(0,255)
ICK00700I DEVICE INFORMATION FOR 2234 IS CURRENTLY AS FOLLOWS:
      PHYSICAL DEVICE = 9336-20
      STORAGE CONTROLLER = 6310
      STORAGE CONTROL DESCRIPTOR = 80
      DEVICE DESCRIPTOR = 10
ICK03020I CPVOL WILL PROCESS 2234 FOR VM/ESA MODE
ICK03090I VOLUME SERIAL = SCRTCH
ICK03011I PAGE RANGE TO BE FORMATTED IS 0 - 255
ICK003D REPLY U TO ALTER VOLUME 2234 CONTENTS, ELSE T
U
ICK03000I CPVOL REPORT FOR 2234 FOLLOWS:

      FORMATTING OF PAGE 0 STARTED AT: 10:32:13
      FORMATTING OF PAGE 255 ENDED AT: 10:32:13
      VOLUME SERIAL NUMBER IS NOW = FB2234

      PAGE ALLOCATION CURRENTLY IS AS FOLLOWS:
      TYPE      START      END      TOTAL
      ----      -
      PERM      4          12499   12496

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
10:32:13      03/16/23

ENTER INPUT COMMAND:
  END

```

The initial allocation of the device is PERM. Pages 0-3 cannot be allocated because they are reserved for the Control Program. The remaining 12,496 pages on the volume are allocated as PERM and correspond to blocks 32–99,999. The range of PERM blocks is calculated as follows:

- There are eight 512-byte blocks per 4096-byte page.
- The initial four pages occupy 32 blocks (4 pages × 8 blocks/page), which are block numbers 0-31. Hence, the first PERM block is at block number 32.
- The device contains a total of 100,000 blocks (12,500 pages × 8 blocks/page), which are block numbers 0-99,999. The last PERM block is 99,999.

The second step allocates space on the volume. In this example, 20 megabytes are allocated for type SPOL and 20 megabytes are allocated for type PAGE on the device that was previously formatted. 20

megabytes is 5,120 pages (40,960 blocks). Space can be allocated by specifying the starting page number and number of pages, but in this example only the amount of space to allocate for each type is specified.

```

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
ENTER ALLOCATION DATA
TYPE PAGES
.....

spol 20m
page 20m
end

HCPCCF6209I INVOKING ICKDSF.
ICK030E DEFINE INPUT DEVICE: FN FT FM, "CONSOLE", OR "READER"
CONSOLE
ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"
CONSOLE
ICKDSF - CMS/XA/ESA DEVICE SUPPORT FACILITIES 17.0
        TIME: 11:01:42      03/16/23      PAGE 1

ENTER INPUT COMMAND:
  CPVOL ALLOC MODE(ESA) UNIT(2234) VFY(FB2234) -
ENTER INPUT COMMAND:
  TYPE((PERM,0,31) -
ENTER INPUT COMMAND:
  (SPOL,32,5151) -
ENTER INPUT COMMAND:
  (PAGE,5152,10271))
ICK00700I DEVICE INFORMATION FOR 2234 IS CURRENTLY AS FOLLOWS:
  PHYSICAL DEVICE = 9336-20
  STORAGE CONTROLLER = 6310
  STORAGE CONTROL DESCRIPTOR = 80
  DEVICE DESCRIPTOR = 10
ICK03020I CPVOL WILL PROCESS 2234 FOR VM/ESA MODE
ICK03090I VOLUME SERIAL = FB2234
ICK003D REPLY U TO ALTER VOLUME 2234 CONTENTS, ELSE T
U
ICK03000I CPVOL REPORT FOR 2234 FOLLOWS:

        PAGE ALLOCATION CURRENTLY IS AS FOLLOWS:
        TYPE      START      END      TOTAL
        ----      -
        PERM      4          31          28
        SPOL      32          5151       5120
        PAGE      5152       10271      5120
        PERM      10272      12499      2228
ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
        11:01:42      03/16/23

ENTER INPUT COMMAND:
  END

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0

```

The PAGE ALLOCATION table shows the following results:

- From the formatting step, the first four pages (0-3) are reserved for the system.
- From the allocation step:
  1. Pages 4-31 are reserved as PERM and can be used for the Stand-Alone Program Loader.
  2. SPOL space occupies pages 32-5,151 (blocks 256-41,215).
  3. PAGE space is next to SPOL space and occupies pages 5,152-10,271 (blocks 41,216-82,175).
  4. The remaining space (pages 10,272-12,499) is allocated as PERM.

**Note:** Pages, blocks, and bytes are related in the following ways:

- A page contains eight blocks.
- A block contains 512 bytes.
- A page contains 4,096 bytes.

- The first block of page N is block number  $(8 \times N)$ . For example, the first block of page 5,151 is  $(8 \times 5,151) = 42,208$ .
- The last block of page N is block number  $(8 \times N + 7)$ . For example, the last block of page 5,151 is  $(8 \times 5,151 + 7) = 42,215$ .
- A graphic example of page 0:

Page 0							
Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7
Bytes 0 - 511	Bytes 512 - 1023	Bytes 1024 - 1535	Bytes 1536 - 2047	Bytes 2048 - 2559	Bytes 2560 - 3071	Bytes 3072 - 3583	Bytes 3584 - 4095

**Example 3: Format and allocate an FBA device by specifying allocation sizes. Request more space than the volume capacity.**

The following example formats and allocates PAGE space on a 1021.82 GB FBA volume. The FBA volume can be a VDISK, EDEVICE, or real FBA device. Although the requested space is greater than the volume's capacity, CPFMTXA adjusts and allocates to the volume's limit.

CPFMTXA generates some input for ICKDSF. In the example screens, user input is in italic text.

*cpfmtxa*

ENTER FORMAT, ALLOCATE, LABEL, OWNER, ERASE OR QUIT:

*f*

ENTER THE VDEV TO BE PROCESSED OR QUIT:

*f000*

ENTER THE PAGE RANGE TO BE FORMATTED ON DISK F000 OR QUIT:

*0-1023*

ENTER THE VOLUME LABEL FOR DISK F000:

*knvm00*

CPFMTXA:  
 FORMAT WILL ERASE PAGES 000000000-000001023 ON DISK F000  
 DO YOU WANT TO CONTINUE? (YES | NO)

*yes*

HCPCCF6209I INVOKING ICKDSF.  
 ICK030E DEFINE INPUT DEVICE: FN FT FM, "CONSOLE", OR "READER"  
 CONSOLE  
 ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"  
 CONSOLE  
 ICKDSF - CMS/XA/ESA DEVICE SUPPORT FACILITIES 17.0  
 TIME: 10:21:57 03/23/23 PAGE 1

ENTER INPUT COMMAND:  
 CPVOL FMT MODE(ESA) UNIT(F000) VOLID(KNVM00) NOVFY -  
 ENTER INPUT COMMAND:  
 RANGE(0,1023)

ICK00700I DEVICE INFORMATION FOR F000 IS CURRENTLY AS FOLLOWS:  
 PHYSICAL DEVICE = 9336-20  
 STORAGE CONTROLLER = 6310  
 STORAGE CONTROL DESCRIPTOR = 80  
 DEVICE DESCRIPTOR = 10

ICK03020I CPVOL WILL PROCESS F000 FOR VM/ESA MODE  
 ICK03090I VOLUME SERIAL = KNVM00  
 ICK03011I PAGE RANGE TO BE FORMATTED IS 0 - 1023  
 ICK003D REPLY U TO ALTER VOLUME F000 CONTENTS, ELSE T  
 U

ICK03000I CPVOL REPORT FOR F000 FOLLOWS:

FORMATTING OF PAGE 0 STARTED AT: 10:21:57  
 FORMATTING OF PAGE 1023 ENDED AT: 10:21:57

VOLUME SERIAL NUMBER IS NOW = KNVM00

PAGE ALLOCATION CURRENTLY IS AS FOLLOWS:  

TYPE	START	END	TOTAL
----	-----	---	-----
PERM	4	267864355	267864352

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0  
 10:21:57 03/23/23

ENTER INPUT COMMAND:  
 END

```

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
ENTER ALLOCATION DATA
TYPE PAGES
.....
page 1024g
end

HCPCCF6209I INVOKING ICKDSF.
ICK030E DEFINE INPUT  DEVICE: FN FT FM, "CONSOLE", OR "READER"
CONSOLE
ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE", OR "PRINTER"
CONSOLE
ICKDSF - CMS/XA/ESA DEVICE SUPPORT FACILITIES 17.0
        TIME: 10:22:08          03/23/23          PAGE  1

ENTER INPUT COMMAND:
  CPVOL ALLOC MODE(ESA) UNIT(F000) VFY(KNVM00) -
ENTER INPUT COMMAND:
  TYPE((PERM,0,31) -
ENTER INPUT COMMAND:
  (PAGE,32,261407))
ICK00700I DEVICE INFORMATION FOR F000 IS CURRENTLY AS FOLLOWS:
  PHYSICAL DEVICE = 9336-20
  STORAGE CONTROLLER = 6310
  STORAGE CONTROL DESCRIPTOR = 80
  DEVICE DESCRIPTOR = 10
ICK03020I CPVOL WILL PROCESS F000 FOR VM/ESA MODE
ICK03090I VOLUME SERIAL = KNVM00
ICK0030D REPLY U TO ALTER VOLUME F000 CONTENTS, ELSE T
U
ICK03000I CPVOL REPORT FOR F000 FOLLOWS:

        PAGE ALLOCATION CURRENTLY IS AS FOLLOWS:
        TYPE      START      END      TOTAL
        ----      -
        PERM      4          31          28
        PAGE      32          267864355  267864324

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
        10:22:12          03/23/23

ENTER INPUT COMMAND:
  END

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
Ready; T=0.01/0.01 10:22:12

```

The PAGE ALLOCATION table shows the following results:

- From the formatting step, the first four pages (0-3) are reserved for the system.
- From the allocation step:
  - Pages 4-31 are reserved as PERM and can be used for the Stand-Alone Program Loader.
  - PAGE space occupies pages 32-267,864,355. The total PAGE space is 267,864,324 pages, which is roughly 1021.82 GB. Although 1024 GB of PAGE space was requested, the volume capacity limited the allocation. CPFMTXA extended the PAGE space to the end of the volume. The same space would be allocated if the user requested 1 TB or a larger value.
- All space definitions combine for a total of 267,864,356 pages (2,142,914,848 blocks) on the volume:
  - Reserved for the system (four pages)
  - Reserved as PERM, perhaps for the Stand-Alone Program Loader (28 pages)
  - PAGE space (267,864,324 pages)
- The 267,864,356 pages contain 2,142,914,848 blocks (267,864,356 pages × 8 blocks/page).

The QUERY EDEVICE command shows the following details for the FBA volume:



**q edev f000 details**

```

EDEV F000 TYPE FBA ATTRIBUTES NVMe
VENDOR: KCM61RUL PRODUCT: 30T7 REVISION:
BLOCKSIZE: 512 NUMBER OF BLOCKS: 2142914848
SEGMENT: 01/28
HYPERPAV POOL: 0
PCIFUNCTION: 00000072
EQID: 8CE38E00F00000000000000000000000F0F0000000007FBA4923
SERIAL NUMBER: Z1N0A004T2K8

```

**Usage Notes**

1. By default, the CPFMTXA utility is on the cross-release utilities disk (PMAINT 551). This disk must be accessed to run the CPFMTXA utility.
2. If ICKDSF is available to the user, it is started to process the function designated by the CPFMTXA parameters. The message INVOKING ICKDSF . is issued to the console. If ICKDSF is not on an accessed disk, no formatting, allocating, or labeling occurs. See [Response 2](#).

Using ICKDSF is the recommended way to format, label, and allocate DASD volumes for CP use. For information about using ICKDSF, see [Device Support Facilities \(ICKDSF\): User's Guide and Reference](#).

3. You can interrupt CPFMTXA processing by pressing the PA1 key.
4. If you do not format an ECKD volume before you use the ALLOCATE option, then a permanent I/O error message is displayed.

When reallocating cylinders or pages from TDSK, PERM, or PARM to PAGE, SPOL, or DRCT, you must reformat these cylinders or pages first. For ECKD devices, for SPOOL and DIRECTORY volume types, extents are limited to cylinders 0-65519, and for PAGE volume types extents are limited to cylinders 0-1182005. TDSK and PARM disks can be located anywhere on the volume, but the PARM disk size is limited to 65520 cylinders. PERM space allocation is allowed anywhere on the volume.

5. If the starting cylinder is 0, CPFMTXA helps to prevent you from accidentally erasing a disk. CPFMTXA indicates the cylinders or pages that are to be formatted, and prompts you to continue. If the cylinders or pages are incorrect, you can reply NO, and the disk remains unchanged. If they are correct, you can reply YES, and processing continues.

If the starting cylinder is not 0, CPFMTXA first verifies the DASD and then prompts you with the cylinder range to be formatted. When formatting a portion of a DASD (anything other than cylinder 0) you must specify the existing label for that DASD. You cannot change the label during the formatting of a DASD unless cylinder 0 is included in the range of cylinders to format.

6. If you enter the CPFMTXA command without the LABEL or ALLOCATE options, it is assumed that you want to format the entire DASD.

For example, suppose you enter `CPFMTXA 192 TEST1`

CPFMTXA assumes that you want to format the entire device. All cylinders or pages are formatted, and all cylinders or pages are initially allocated as PERM space. Then, you are prompted to update the DASD allocation, and the volume on 192 is labeled TEST1.

However, suppose that you enter `cpfmtxa 192 test2 0.3`

CPFMTXA formats cylinders or pages 0, 1, and 2. Since you formatted cylinder or page 0, all cylinders or pages are to be initially allocated as PERM space. Then, you are prompted to update the DASD allocation, and the volume on 192 is labeled TEST2.

If you enter `cpfmtxa 192 test3 1-3`, then CPFMTXA first checks to see whether the volume ID on 192 is TEST3. If it is not, you receive an error message that indicates what the correct label is, and CPFMTXA processing ends. If *valid* is TEST3, CPFMTXA formats cylinders or pages 1, 2, and 3, and you are prompted to update the DASD allocation. Since you did not format cylinder or page 0, the label is not written, and the cylinders or pages are not initially allocated as PERM.

7. DRCT, PAGE, PARM, PERM, SPOL, and TDSK are allocation types. You can use these entries in any order. Each entry in turn overlays the allocation record, and any space that is not reallocated remains the same.

**Note:** Special care must be exercised in this area, especially when cylinder 0 or pages 0-3 are reformatted. For example, if new directory cylinders or pages are allocated, then the allocation record is overwritten, effectively destroying any pointers to the active directory and rendering it inaccessible. If an allocation entry overlays a previous cylinder or page allotment, the previous cylinder or page space allotment is truncated to the beginning of the next cylinder or page allotment. For example, assume that the following example is the current allocation of a vdev:

CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:			
TYPE	START	END	TOTAL
PERM	0	0	1
PARM	1	29	29
PERM	30	554	525

Assume that you enter the following allocation data:

```
parm 20-49
tdsk 50-100
drct 100-200
spol 150-500
page 400.50
```

The actual output follows:

CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:			
TYPE	START	END	TOTAL
PERM	0	0	1
????	1	19	19
PARM	20	49	30
TDSK	50	99	50
DRCT	100	149	50
SPOL	150	399	250
PAGE	400	449	50
SPOL	450	500	51
PERM	501	554	54

**Note:** A PARM range must be entirely reallocated when a change is made. If the PARM range is not entirely reallocated, the ranges that are not reallocated are set to indicate that undefined allocation types exist.

- Space for system dumps and system data files must be allocated as SPOL. For a CKD/ECKD device, you must not allocate cylinder 0 as TDSK. For an FBA device that is formatted for z/VM, ICKDSF reserves pages 0-3 (blocks 0-31) for system use.

If the SALIPL utility is then used to install Stand-Alone Program Loader (SAPL) onto an FBA device, then pages 0-25 (blocks 0-207) are reserved for system use. These reserved FBA pages must be allocated only as PERM and must not later be provisioned as user minidisks.

- These types of allocations cannot reside above page 16,777,214:
  - DRCT allocations on FBA devices and EDEVICES
  - SPOL allocations on FBA devices and EDEVICES
  - PAGE allocations on FBA devices and EDEVICES, except for NVMe EDEVICES

For PAGE allocations on NVMe EDEVICES, the maximum block number is limited only by the device size.

All other FBA allocations (PERM, PARM, TDSK) can reside above page 16,777,214.

Message HCP381E is issued if allocation is attempted of a higher page than is allowed.

- For ECKD devices, for SPOOL, PARM, and DIRECTORY volume types, extents are limited to cylinders 0-65519, and for PAGE volume types extents are limited to cylinders 0-1182005. TEMP and PARM disks can be located anywhere on the volume, but the PARM disk size is limited to 65520 cylinders. PERM space allocation is allowed anywhere on the volume. Message HCP138E or HCP139E is issued if you attempt to define a user volume of type other than PERM above 65519 cylinders.

11. If you do not format a volume before you use the OWNER option, a permanent I/O error message is displayed.

When a volume is initially formatted, the ownership information is set to NOSSI and NOSYS. You must then issue CPFMTXA with the OWNER option to set the appropriate ownership information.

12. Ownership information is written to cylinder 0 and must be assigned only to a volume that is CP-formatted. Reformatting cylinder 0 destroys any ownership information that is recorded there.
13. For information about how DASD ownership information is used when the CP-owned list is processed, see [How the CP-Owned List Is Processed](#) in *z/VM: CP Planning and Administration*.

## Responses

Response 1:

```
INVOKING ICKDSF.
```

This response is issued to alert the user that ICKDSF is used instead of CPFMTXA to format, label, and allocate DASD volumes.

Response 2:

```
DEVICE SUPPORT FACILITIES (5684-042) IS NECESSARY FOR
CP VOLUME MAINTENANCE. FILE ICKDSF MODULE CANNOT BE
FOUND ON ANY ACCESSED DISK; CPFMTXA PROCESSING ENDED.
```

This response is issued to alert the user to access ICKDSF. CPFMTXA translates user input only to input that is acceptable to ICKDSF.

For information on using ICKDSF to format, label, and allocate DASD volumes, see [Device Support Facilities \(ICKDSF\): User's Guide and Reference](#).

Response 3:

```
ENTER FORMAT, ALLOCATE, LABEL, OWNER OR QUIT:
```

This prompt is issued if you do not supply both *vdev* and *valid* on the command line when CPFMTXA is started and you want to change ownership of the CP-formatted device.

### FORMAT

indicates that you want to format the disk. If you format the disk from cylinder or page 0, you are prompted for a volume label to be written on the disk. If you are not formatting cylinder or page 0, the volume label you enter is verified with the existing volume label of the disk. After the disk is formatted, you are prompted to enter the disk allocations.

### ALLOCATE

indicates that you want to change only the present allocations. The ALLOCATE option requires that the disk is already formatted.

### LABEL

indicates that you want to change only the volume label on the disk.

### OWNER

indicates that you want to change the ownership information. Ownership information consists of an SSI cluster name and a system name.

### QUIT

indicates that you want to end CPFMTXA.

Any truncation of FORMAT, ALLOCATE, LABEL, OWNER, or QUIT is acceptable. For example, you can enter FORMAT as F, FO, FOR, FORM, FORMA, or FORMAT.

Response 4:

```
ENTER THE VDEV TO BE PROCESSED OR QUIT:
```

This prompt is issued when *vdev* is not entered on the command line when you start CPFMTXA.

**VDEV**

is the virtual device number of the disk to be formatted, allocated, or labeled. Any truncation of QUIT is acceptable (Q, QU, QUI, or QUIT).

Response 5:

```
ENTER THE {CYLINDER|PAGE} RANGE TO BE
FORMATTED ON DISK vdev OR QUIT:
```

This prompt is issued when FORMAT is selected in Response 3.

**vdev**

is the virtual device number that was either specified on the command line or entered in response to a prompt.

Response 6:

```
ENTER THE VOLUME LABEL FOR DISK vdev:
```

**vdev**

is the virtual device number that was either specified on the command line or entered in response to a prompt. This prompt is issued when *valid* is not entered on the command with which you start CPFMTXA.

Response 7:

```
FORMAT WILL ERASE {CYLINDERS|PAGES} range ON DISK vdev
DO YOU WANT TO CONTINUE? (YES | NO)
```

This prompt is issued whenever you format a disk. If you reply YES, ICKDSF is started and the format function continues. If you reply NO, CPFMTXA processing ends.

**range**

is inclusive, and specifies the numbers of the cylinders or pages to be formatted.

**vdev**

is the virtual DASD to be formatted.

Response 8:

```
ENTER ALLOCATION DATA
TYPE {CYLINDERS|PAGES}
.....
```

This response is issued whenever you update the allocation map. The response prompts you to update the allocation map. Enter your allocation data from the console; this data must contain both (1) the allocation type, and (2) the cylinders or pages to be allocated as that type. When you enter END, the updated allocation map is displayed on your console, by using the CURRENT ALLOCATION response. For more information, see [Changing the Allocation Map](#).

**type**

can be DRCT, PAGE, PARM, PERM, SPOL, or TDSK.

**nnnnnn-nnnnnn**

is the cylinder or page range for the corresponding type.

Response 9:

```
REENTER: TYPE AND {CYLINDERS|PAGES} OR 'END'
```

This prompt is issued whenever you enter erroneous allocation data. An error message that explains the error precedes this prompt. You are prompted to correct and reenter the line of allocation data that contained the error.

Response 10:

```
ERASE WILL DESTROY THE CONTENTS OF DISK vdev ON MDISK ownerid ovdev
DO YOU WANT TO CONTINUE? (YES | NO)
```

This prompt is issued whenever you erase a disk by using the fast erase function. If you reply YES, processing continues and the following information is displayed:

```
Erasing vdev with      NNNNN cylinders at HH:MM:SS
Erased   AAAA cylinders at: HH:MM:SS
Erased   BBBB cylinders at: HH:MM:SS
```

If you reply NO, CPFMTXA processing ends.

***vdev***

the virtual device number of the DASD to be erased.

***ownerid***

the user ID of the owner of the virtual device number of the DASD to be erased.

***ovdev***

the owner's virtual device number of the DASD to be erased.

***NNNNN***

is the number of cylinders to be erased on the virtual device.

***HH:MM:SS***

is the time at which the erase function was completed.

***AAAA***

***BBBB***

is the number of cylinders that are deleted since the ERASE command started.

## Messages

**Note:** The messages that are returned by the CPFMTXA utility are CP messages.

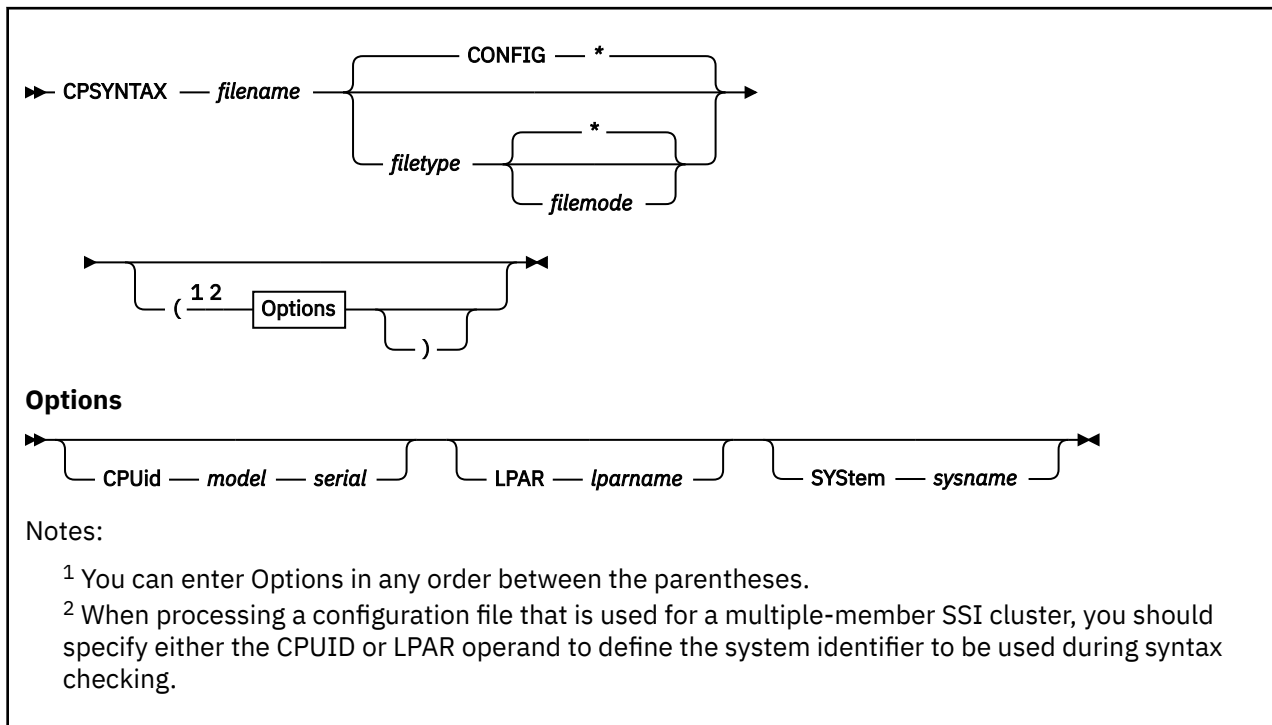
Messages:

- HCP003E INVALID OPTION - *option*
- HCP006E INVALID DEVICE TYPE - *vdev*
- HCP022E A VIRTUAL DEVICE NUMBER WAS NOT SUPPLIED OR IT WAS INVALID.
- HCP023E VOLID MISSING OR INVALID
- HCP024E {CYLINDERS|PAGES} MISSING OR INVALID
- HCP040E DEVICE *vdev/rdev/ldev* DOES NOT EXIST
- HCP263E TOO MANY OPERANDS WERE SUPPLIED FOR THIS COMMAND
- HCP380E INVALID {CYLINDER|PAGE} TYPE - '*type*'.
- HCP381E INVALID {CYLINDER|PAGE} RANGE - '*range*'.
- HCP382E EXTRANEIOUS DATA DETECTED - '*data*'.
- HCP386E {FORMATTING|ALLOCATING|ERASING} CANNOT BE DONE ON DASD *vdev* BECAUSE IT IS READ-ONLY.
- HCP389E THE {CYLINDER|PAGE} RANGE IS TOO LARGE BECAUSE DEVICE *vdev* HAS *nnnnnn* {CYLINDERS|PAGES}
- HCP399E TDSK CANNOT BE ALLOCATED ON CYLINDER 0
- HCP6209I INVOKING ICKDSF.
- HCP6210I DEVICE SUPPORT FACILITIES (5684-042) IS NECESSARY FOR CP VOLUME MAINTENANCE. FILE ICKDSF MODULE CANNOT BE FOUND ON ANY ACCESSED DISK; CPFMTXA PROCESSING ENDED.
- HCP6211E HCPFER IS NECESSARY FOR THE ERASE FUNCTION. FILE HCPFER MODULE CANNOT BE FOUND.
- HCP6212E UNKNOWN RETURN CODE FROM HCPFER MODULE.

- HCP6213E THE DEVICE SPECIFIED DOES NOT SUPPORT THE ERASE FUNCTION.
- HCP6628E OWNERSHIP NAME MISSING OR INVALID.
- HCP6632E THE \* OPERAND IS NOT VALID WHEN THE SPECIFIED SSI NAME DIFFERS FROM THE ACTUAL SSI NAME.
- HCP6633E THE SPECIFIED COMBINATION OF OPERANDS IS ONLY VALID WHEN NOT CONFIGURED FOR SSI.
- HCP6639E UNABLE TO DETERMINE THE SSI CLUSTER NAME. ISSUE THE COMMAND AGAIN WITH CLUSTER NAME SPECIFIED.
- HCP8612E I/O ERROR OCCURRED ON DIAGNOSE X'A8' DURING FAST ERASE FUNCTION.

For information about ICKDSF messages and responses, see [Device Support Facilities \(ICKDSF\): User's Guide and Reference](#).

## CPSYNTAX



### Purpose

Use the CPSYNTAX utility to check the syntax of the system configuration file.

CPSYNTAX catches incorrect and unrecognized statements in the system configuration file. It does not attempt to identify problems arising between statements or valid but duplicate operands on a single statement. CPSYNTAX does not guarantee that it will find all configuration file changes that will lead to a problem during system IPL.

During system IPL, configuration file postprocessing routines do further checking of the data specified in the configuration file. IPLing a second-level system to check the configuration file is recommended for a more thorough test and might find problems that CPSYNTAX does not. However, second-level IPL does not eliminate environmental factors of the real production environment and still might not find all problems related to the system configuration file.

Some items that CPSYNTAX does not address:

- Ensuring that some CP-owned volume remains STARTed for PAGE (DRAIN data).
- Ensuring that some operator console remains available (OFFLINE\_AT\_IPL).
- Inappropriate positioning of SYSTEM\_IDENTIFIER statements.
- Inappropriate ordering of DRAIN/CP\_OWNED statements or of TIMEZONE\_DEFINITION/TIMEZONE\_BOUNDARY statements.

### Operands

#### **filename**

is the file name of the configuration file you want to be syntax checked.

#### **filetype**

is the file type of the configuration file you want to be syntax checked. The default file type is CONFIG.

#### **filemode**

is the file mode of the configuration file you want to be syntax checked. The default file mode is \*.

## Options

### CPUI*d model serial*

where *model* is the CPU model, such as 2097, and *serial* is the CPU serial number, up to six hexadecimal digits, of the processor on which you eventually want the file to be processed. The CPUID option can be used to pass environmental data to CPSYNTAX to cause SYSTEM\_IDENTIFIER statements to match.

### LPAR *lparname*

where *lparname* is the 1- to 8-character logical partition name on which you eventually want the file to be processed. The LPAR option can be used to pass environmental data to CPSYNTAX to cause SYSTEM\_IDENTIFIER statements to match.

### SYStem *sysname*

where *sysname* is the 1- to 8-character string that will be used as the default system identifier. The SYSTEM option causes SYSTEM\_IDENTIFIER\_DEFAULT statements to be ignored. If a matching SYSTEM\_IDENTIFIER statement is found, the *sysname* specified on the matching statement becomes the new system identifier.

## Usage Notes

1. Imbedded files are processed by CPSYNTAX, but all imbedded files must reside on the same minidisk as the base configuration file.
2. By default, the CPSYNTAX utility is located on the system tools disk (MAINT 193). The utility consists of an exec called CPSYNTAX and a module called CPSYNCHK. The CPSYNTAX EXEC is a REXX exec that can be compiled by the REXX compiler.
3. The CPUID and LPAR options can be used to pass environmental data to CPSYNTAX to cause SYSTEM\_IDENTIFIER statements to match. The options CPUID, LPAR, and SYSTEM can be specified on the same CPSYNTAX invocation.
4. If your configuration file has a statement that contains the string &LPARNAME, or a SYSTEM\_IDENTIFIER statement that specifies the LPAR operand, you must use the CPSYNTAX LPAR option. If you do not, those statements will be flagged as an error, because the LPAR name cannot be determined.

## Messages

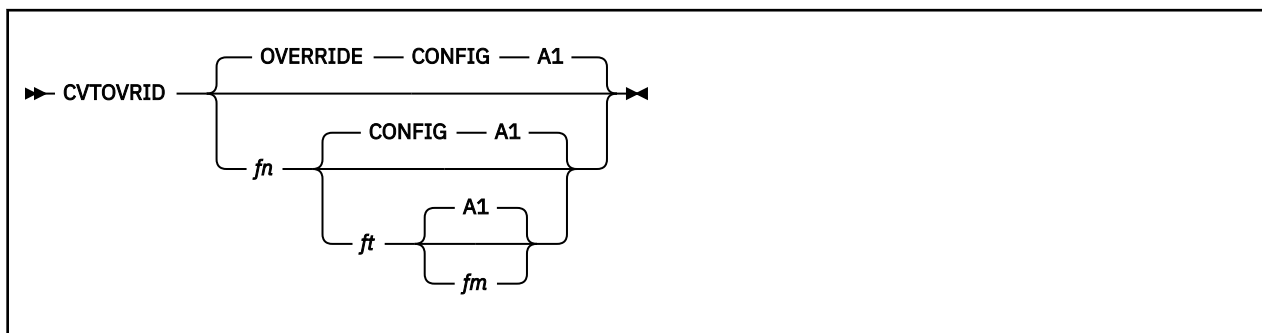
- HCP002E Invalid operand - *operand*
- HCP003E Invalid option - {command|statement} contains extra option(s) starting with *option*
- HCP003E Invalid option - *option*
- HCP013E Conflicting option - *option*
- HCP1001E An operand is missing for *option*.
- HCP1646E Too many qualifiers specified on BEGIN or END statement. No more than 64 qualifiers may be specified.
- HCP1647E Unexpected END statement found. BEGIN and corresponding END statement must be in the same file.
- HCP1671E This system is not a member of the SSI cluster.
- HCP1672E SSI statement encountered but system identifier has not been defined
- HCP1672E The system ID was changed by a SYSTEM\_IDENTIFIER statement after the SSI statement was processed.
- HCP1673E The system identifier *sysname* is not the same as the gateway identifier *gateid*. They must be the same because this system is a member of an SSI cluster.
- HCP1675E Missing qualifier on BEGIN statement
- HCP1676E No END statement found for BEGIN statement on line *nnnnnnnn* of file *fn ft*. BEGIN and corresponding END statement must be in same file.



- HCP1677E Unexpected END statement because no BEGIN statement is in effect.
- HCP1678E BEGIN statement encountered while another BEGIN statement is in effect.
- HCP1679E Specified qualifier(s) do not match qualifiers that are in effect.
- HCP1680E Record qualifiers not allowed because BEGIN statement is in effect.
- HCP1681E SSI statement already processed - statement ignored
- HCP1683E *statement* statement on line *number* of file *fn ft fm* must not be specified with SSI statement.
- HCP1685E The same system ID appears more than once on the SSI statement.
- HCP1687E The statement failed because the LPAR name is unknown. Specify the LPAR option of CPSYNTAX.
- HCP1687E &LPARNAME cannot be resolved. Specify the LPAR option of CPSYNTAX.
- HCP1689W An error was found related to a BEGIN or END statement. System IPL will fail unless this error is corrected.
- HCP2149E A variable name starting with reserved prefix "CP." cannot be set.
- HCP2585E Reconfigurable storage size of *stgsize* exceeds the maximum allowed reconfigurable storage size of *maxsize*. Reconfigurable storage is restricted to a maximum of *xx%* of all online storage.
- HCP2768E Missing variable name
- HCP6700E File *fn ft*, [*nnnnnnnn* through] *nnnnnnnn*:
- HCP6701E Invalid {logo|system} configuration file statement - *operand*
- HCP6702E Error encountered while attempting to read records from *fn ft [fm]*
- HCP6703E File *fn ft fm* not found.
- HCP6704E Missing {CPU model number | CPU serial number | system ID | lparname | token} at end of line.
- HCP6705E Comment starting on line *nnnnnnnn* in *fn ft* was never completed.
- HCP6705E Statement starting on line *nnnnnnnn* in *fn ft* was continued past end of file.
- HCP6706E Invalid CPU model number - *string*
- HCP6706E Invalid CPU serial number - *string*
- HCP6706E Invalid file name - *string*
- HCP6706E Invalid file type - *string*
- HCP6706E Invalid identifier - *string*
- HCP6706E Invalid variable name - *name*
- HCP6706E Invalid variable range - *range*
- HCP6707E Statement exceeded maximum length of 4000 on record *nnnnnnnn* in file *fn ft*.
- HCP6709E Too many items specified - *token*
- HCP6710E File *fn ft* does not exist and cannot be imbedded.
- HCP6710E Imbedding file *fn ft* would cause an embed loop -- statement ignored.
- HCP6710E System cannot be resolved because system ID is undefined -- file *fn ft* cannot be imbedded.
- HCP6711E Duplicate volume identifier *nnnnnn* specified -- statement ignored.
- HCP6711E Slot *nnn* already contains [system|volume] *nnnnnn* -- statement ignored.
- HCP6716E Already [not] tolerating configuration file errors -- statement ignored.
- HCP6725E XLINK system ID *sysID* is a duplicate -- statement ignored.
- HCP6820E Required file name has been omitted.
- HCP6821E Invalid quoted string encountered -- statement ignored.
- HCP6822E Statement contained only qualifiers.
- HCP6823E *fn* MODULE was not found.

- HCP6824E Unexpected return code *rc* received from module *fn*.
- HCP6825E Unknown conversion type encountered -- may need a different version of *fn* module.
- HCP6826E One or more errors were encountered in processing sections of the system configuration file that were marked not to tolerate errors.
- HCP6827E XLINK systems were specified and *sysname* was in neither the XLINK system inclusion nor the exclusion list. Failure to correct this error may lead to loss of user data.
- HCP6828E No {CP owned volumes|system residence area| operator consoles} defined in the configuration file.
- HCP6829W Configuration file processing complete -- no errors encountered.

## CVTOVRID



### Purpose

Use the CVTOVRID utility to convert a CMS class override file (formerly processed by the CP OVERRIDE utility) into a series of MODIFY statements to imbed into a system configuration file.

**Note:** CVTOVRID is an XEDIT MACRO, so you must be editing (XEDITing) the file that you want to convert when you invoke this utility.

### Operands

#### *fn*

is the file name of the output file that will contain the MODIFY configuration file statements. If omitted, the default file name is OVERRIDE.

#### *ft*

is the file type of the output file that will contain the MODIFY configuration file statements. If omitted, the default file type is CONFIG.

#### *fm*

is the file mode of the output file that will contain the MODIFY configuration file statements. You must have a link that allows you to write to the disk (or SFS directory) accessed at the specified file mode. If omitted, the default file mode is A1.

### Usage Notes

1. By default, the CVTOVRID utility is located on the system tools disk (MAINT 193).
2. After creating the file of MODIFY statements, you must:
  - a. Add an IMBED statement to your system configuration file to pull in the new file (or move the MODIFY statements from the new file to your system configuration file)
  - b. Make sure that the new file (or your system configuration file) is on the parm disk.
3. CP maintains some of the QUERY subcommand control blocks off of a virtual subcommand chain. These blocks are for QUERY VIRTUAL subcommands and generic subcommands for which there is no specific operand that identifies them. For example, QUERY 0003 for a class B user is identified by the generic QUERY subcommand block because the second operand, which would be used to identify it, is a device address of which there is no single keyword that indicates that a real device is being queried. If no keyword is found, then the entry point for the generic QUERY subcommand is used, provided that the user has the correct privilege class.

CVTOVRID creates MODIFY CMD commands for the virtual QUERY subcommand chain when it determines that the UCR file (that would have been created under User Class Restructure, which is no longer supported) would have affected the subcommand blocks on that chain. This occurs for CP privilege class B and G subcommands.

## Responses

Response 1:

To convert a CMS class override file into a file of MODIFY statements for your system configuration file, enter the following on the command line of your XEDIT screen:

```
====> cvtovrid newstmts config a
```

This causes XEDIT to create a file called NEWSTMTS CONFIG that you can file on your A-disk (or SFS directory). You can then imbed this file in your system configuration file or place the statements in NEWSTMTS CONFIG directly in your system configuration file.

## DDR



### Purpose

Use the DASD Dump Restore (DDR) utility to dump, copy, or print data which resides on z/VM user minidisks or dedicated DASDs. The utility may also be used to restore or copy DASD data which resides on z/VM user tapes. There are two versions — a CMS command (DDR) and an IPLable stand-alone program.

To run the DDR utility as a stand-alone program under CP, you need to use IPL DDRXA. See [z/VM: System Operation](#) for examples.

The DDR utility has five functions:

1. Dumping part or all of the data from a DASD to tape.
2. Transferring data from tapes created by the DDR dump function to a direct access device. The direct access device must be the same type as that which originally contained the data.
3. Copying data from one device to another of the same type. Data may be reordered by cylinder or block when copying it from disk to disk. In order for one tape to be copied to another, the original tape must have been created by either the DDR command or the DDRXA stand-alone utility DUMP function.
4. Printing selected parts of DASD and tape records in hexadecimal and EBCDIC on the virtual printer.
5. Displaying selected parts of DASD and tape records in hexadecimal and EBCDIC on the terminal.

### Operands

*fn ft fm*

\*

the file containing the control statements for the DDR utility. If no file identification is provided, the DDR utility attempts to obtain control statements from the console. The file mode defaults to asterisk (\*) if a value is not provided.

### Usage Notes

1. In the CMS environment, all output is directed (by default) to the CMS printer 00E, unless you specify the SYSPRINT CONS option. Any other SYSPRINT devno option specification is ignored.
2. When you start DDR in CMS in a virtual machine environment, the I/O operation is performed by CP. CP has better error recovery facilities.
3. When performing the DUMP or COPY functions, the stand-alone DDR utility does not support cylinder faults for mass storage subsystem (MSS) virtual volumes.
4. DDR copies mode zero (private) files from a CMS disk, regardless of whether the disk is linked R/O or R/W.
5. Results are unpredictable if you use DDR to copy data from one minidisk to another minidisk unequal in size (for example, the output disk contains fewer cylinders or blocks than the input disk).
6. If the DDR stand-alone utility is IPLed from tape with a LOADPARAM of AUTOxxxx, it automatically restores the tape to the DASD with device address xxxx. No output is produced on the console. Command completion is indicated by loading a disabled wait PSW. A wait code of zero in this PSW

indicates a successful restore; a non-zero wait code signals an unsuccessful restore. See *z/VM: CP Messages and Codes* for PSW wait state details.

7. DDR writes its own tape marks after a dump. It reads past any tape marks found between dumps during a restore.
8. In an emergency that requires data to be restored to a SCSI device in order for recovery to be performed, do the following:
  - IPL the initial install DVD.
  - Define the appropriate EDEVICES.
  - Run DDR under CMS in the MAINT virtual machine.

For more information, see the *z/VM: Installation Guide*.

9. To use DDR as a PIPE stage, the DRPC MODULE must first be run to install the DDR PIPE stage. This installation of the PIPE stage will last until CMS is re-IPLed.

#### DDR Control Statements

DDR control statements describe the intended processing and the needed I/O devices. Specify I/O definition statements first.

All control statements may be entered from either the console or the card reader. The program inspects only columns 1 to 71. It ignores all data after the last operand in a statement. Since the output tape must have the DASD cylinder header records in ascending sequences, you must enter extents in sequence by DASD location (that is, in sequence by cylinder number if this is a count-key-data device). You may specify only one type of function to be performed in one execution. Choose from:

- Dump
- Restore
- Copy

You may enter up to 300 statements describing cylinder or block extents.

The function statements are delimited by an INPUT or OUTPUT statement, or by a null line if the console is used for input. If additional functions are to be performed, the sequence of control cards must be repeated. If you do not use INPUT or OUTPUT control statements to separate the functions you specify, an error message (HCP702E) is displayed when the input is read from a card reader or CMS file. The remainder of the input stream is checked for proper syntax, but no further DDR operations are performed. Only those statements needed to redefine the I/O devices are necessary for subsequent steps. All other I/O definitions remain the same.

**To return to CMS**, enter a null line (carriage return) in response to the prompting message (ENTER:). **To return directly to CP**, enter #CP.

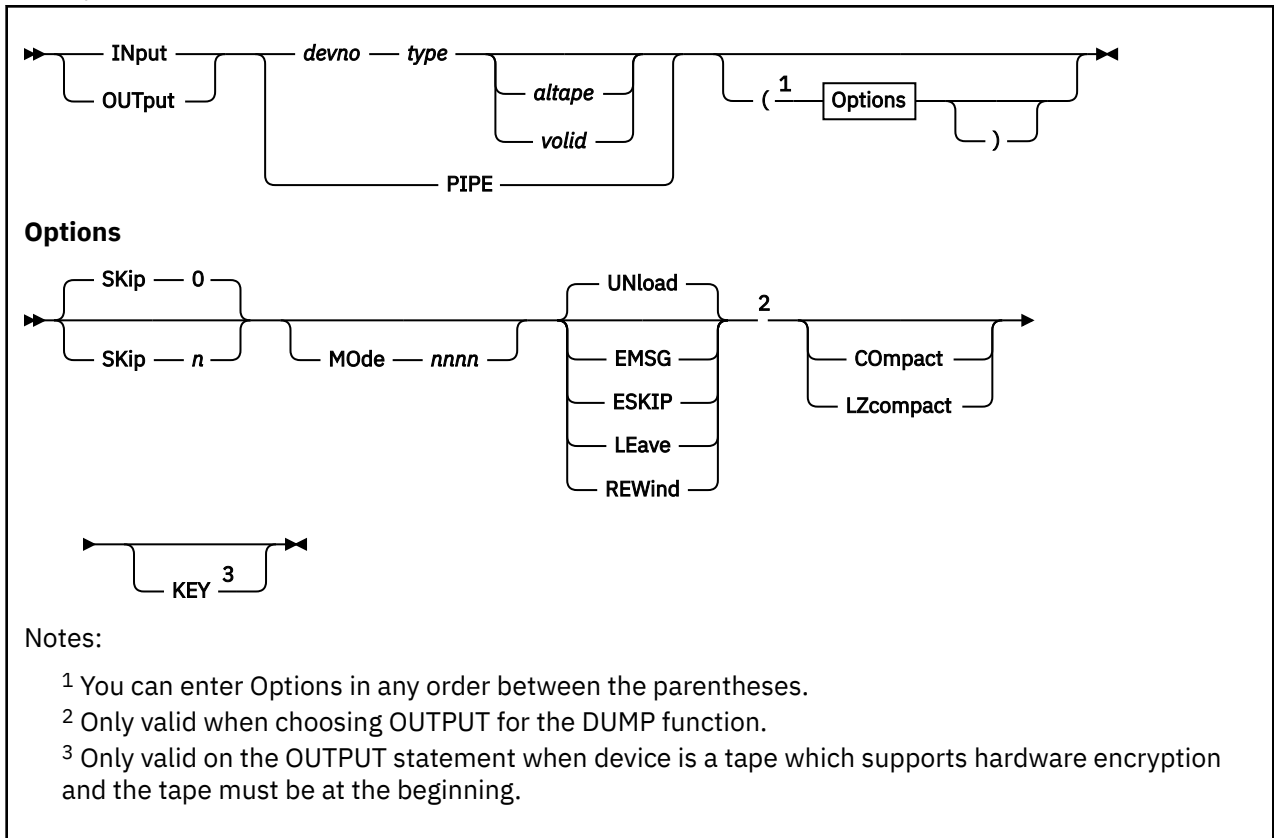
Two logical line editing symbols are recognized and handled by the DDR command. The first is the logical character-delete symbol (@) that allows deletion of one or more of the previous characters entered. The second is the logical-line delete symbol (¢) that deletes the entire previous physical line. Most often, the default values for these two symbols are defined for each virtual machine at system generation. When you use the CP TERMINAL command to redefine the CHARDEL and LINEDEL characters, the redefinitions have no effect on DDR line editing. DDR continues to recognize the at sign (@) and the cent sign (¢) as valid CHARDEL and LINEDEL symbols, respectively.

The PRINT and TYPE statements work differently from other DDR control statements in that they operate on only one data extent at a time. If the input is from a tape created by the dump function, it must be positioned at the header record for each step. The PRINT and TYPE statements have an implied output of either the console (TYPE) or system printer (PRINT). Therefore, PRINT and TYPE statements need not be delimited by an INPUT or OUTPUT statement.

#### I/O Definition Control Statements

The I/O definition statements describe the tape, DASD, and printer devices used while the DASD Dump Restore utility is running.

INPUT/OUTPUT Control Statement



An INPUT or OUTPUT statement describes each tape and DASD unit used.

**INput**

the device described is an input device.

**OUTput**

the device described is an output device.

**Note:** If the output device is a DASD and DDR is running under CMS, the device is released using the CMS RELEASE command function, unless the device is the CMS system disk.

**devno**

the device number of the input or output device.

**type**

the device type. The valid device types are:

DASD	Tape
3380	3422
9336	3480
3390	3480
FB-512	3490
DASD	3590
	Tape

**Notes:**

1. DDR does not provide 7-track support for any tape devices.

2. Specify 3380 for a 3390 that is in 3380 track compatibility mode. Specify 3390 for any 3390 operating in native mode.
3. Any fixed-block device may generically be specified by typing "FB-512" (minimum abbreviation is "FB").
4. The DDR utility, executing in a virtual machine, uses I/O DIAGNOSE X'A8' to perform I/O operations on tape and DASD. DDR under CMS requires that the device type entered agree with the device type of the real device as recognized by z/VM. If there is a conflict with device types, the following message is issued:

```
HCP708E INVALID INPUT OR OUTPUT DEFINITION
```

5. If you enter a specific device type on the INPUT or OUTPUT statement that does not match the actual type of the device that the statement is identifying, then DDR issues the following message:

```
HCP708E INVALID INPUT OR OUTPUT DEFINITION
```

### **valid**

the volume identifier of the disk originally dumped to tape. If the keyword SCRATCH is specified instead of the volume identifier, no label verification is performed.

### **altape**

the address of an alternate tape drive. The alternate tape address is only used when a DUMP command reaches the end of the tape, or a read operation reaches the END-OF-VOLUME marker placed at the end of the tape during a DUMP operation.

**Note:** If multiple reels of tape are required and *altape* is not specified, DDR types the following at the end of the reel:

```
END OF VOLUME CYL xxxxxxxx HD xx, MOUNT NEXT TAPE
```

or

```
END OF VOLUME BLOCK xxxxxx, MOUNT NEXT TAPE
```

After the new tape is mounted, DDR continues automatically.

### **PIPE**

specifies that rather than a device, DDR will use CMS PIPES for input and output. This operand is meant to be used when invoking DDR as a PIPE stage. Note that the DRPC MODULE must first be executed to install the DDR PIPE stage.

*Example 1: Dumping a disk to a CMS file via PIPES*

```
Control Statement File DUMP191 DDR L:
INPUT 191 DASD
OUTPUT PIPE ( LZ
PROMPTS OFF
DUMP ALL

PIPE DDR DUMP191 DDR L | > 191 DUMP L
```

The result will be a file 191 DUMP L that contains the contents of the 191 disk.

*Example 2: Restoring a previously dumped disk via PIPES*

```
Control Statement File RES191 DDR L:
INPUT PIPE ( LZ
OUTPUT 191 DASD
PROMPTS OFF
RESTORE ALL

PIPE < 191 DUMP L | DDR RES191 DDR L
```

This will result in the contents of the file 191 DUMP L being written to the 191 disk.



**SKip *n*****SKip 0**

forward spaces *n* files on the tape, *n* is any number up to 65535. (See note “6” on page 2153.) The SKIP option is reset to zero after the tape has been positioned. The zero value is the default.

**MOde *nnnn***

selects the recording format in which data will be written on the tape. On 3422 tape devices, you can specify the recording format (the MODE) only at the beginning of a tape volume. No changes in the format can be made after this initial designation. On 3480 and 3490 (including 3490Es), you can change the recording format (the MODE) each time you choose to write to a tape volume. On 3590 tape devices you can only change device independent mode option values each time you choose to write to a tape volume, while device dependent mode option values can only be specified at the beginning of a tape volume (see note “8” on page 2153. After you write to a volume, the MODE value returns to the default value for that particular tape device. If you wish to write to the tape in a different recording format next time, you must again specify the MODE value.

The table below specifies the valid and default mode option values for supported tape devices:

Device Type	Valid Device-Dependent (MODE <i>nnnn</i> ) Option Values	Valid Device-Independent (MODE <i>nnnn</i> ) Option Values	Device (MODE <i>nnnn</i> ) Default Value
3422	1600, 6250	n/a	1600
3480	38K	NOCOMP	38K
3480 w/IDRC	XF, 38K	COMP, NOCOMP	38K
3490	3490C, 3490B	COMP, NOCOMP	3490C
3590	EFMT1, EFMT2, EFMT3, EFMT4	COMP, NOCOMP	COMP

**Notes:**

1. If you specify a density mode that the tape cannot handle, the control unit may not return an error condition. In this case, the mode setting is ignored, and the default control unit setting is used.
2. The 3490 tape devices, models A01, A02, B02, B04, D31, and D32, should be treated as type 3480 devices. All other 3490 device models should be treated as type 3490 devices. For a complete breakdown of 3480 and 3490 device types by model number, see [Defining I/O Devices in z/VM: CP Planning and Administration](#).
3. COMP and NOCOMP are device-independent MODE options for compaction and non-compaction, respectively, for use with either 3480, 3490, or 3590 tape subsystems.
4. On 3480 tape drives only, if you specify both hardware (MODE XF or COMP) and software (COMPACT) compaction options, the hardware support takes precedence.  
  
For 3490s, where hardware compaction is the default (MODE 3490C or MODE COMP), software compaction will take precedence, if specified.  
  
It is recommended that both software compaction and hardware compaction not be specified for any tape device.
5. Data is always translated into non-compacted form when read from tape.
6. All other tape related commands view a file as a CMS file located on a minidisk. However, DDR's definition of a file is anything that is dumped, restored, or copied by one DDR command. There could be hundreds of CMS files in just one DDR file.
7. By specifying an EFMTx mode explicitly, the device default COMP value is also active.
8. For specifying an EFMTx mode, the tape must be at the beginning. If data has already been written to the tape using a specific mode and/or the tape is not at the beginning (for example, SKIP option), any further data dumped to the tape will also use the already, formerly used mode.

**REWind**

rewinds the tape when a function completes.

**UNload**

rewinds and unloads the tape when a function completes. This is the default.

**LEave**

leaves the tape positioned at the end of the file when a function completes.

**EMSG**

allows the user to determine whether to continue after an I/O error on the INPUT device. The user must respond to the prompt 'DO YOU WISH TO CONTINUE? RESPOND YES OR NO:' which will be issued with messages HCP705D and HCP714D. This option may only be specified on the INPUT statement.

**ESKIP**

allows the user to continue automatically after an I/O error on the INPUT device without having to supply a 'YES' to the prompt 'DO YOU WISH TO CONTINUE? RESPOND YES OR NO:'. This option may only be specified on the INPUT statement.

**COmpact**

causes the output tape to be in a compact format. Output in compact format uses less tape space than output in standard format. DDR stores data in a compact format by compressing strings of duplicate data into a smaller amount of space, thus reducing the amount of space necessary to represent the characters in the data.

This option is valid only on the OUTPUT control statement for the DUMP function. Tapes in the compact format may be used as input to the RESTORE, COPY, PRINT, and TYPE functions without any changes. Refer to the following sections on function statements for more information.

You cannot use tapes created with the COMPACT option as input to earlier levels of DDR or IPL DDRXA that do not support the compaction function.

**LZcompact**

causes the output tape to be in a compact format. Output in a compact format uses less tape space than output in standard format. DDR stores data in a compact format by using the processor's compression call instruction to compress the data before writing it out to tape.

This option is valid only on the OUTPUT control statement for the DUMP function. Tapes in the compact format may be used as input to the RESTORE, COPY, PRINT, and TYPE functions without any changes. Refer to the following sections on function statements for more information.

You cannot use tapes created with the LZCOMPACT option as input to earlier levels of DDR or IPL DDRXA that do not support the LZcompact function.

**KEY**

causes encryption to be used for the output type. If the LABEL/HASH control statement is also specified, the label value specified will be the key label used for tape encryption. Otherwise, default key encryption will be used.

**Notes:**

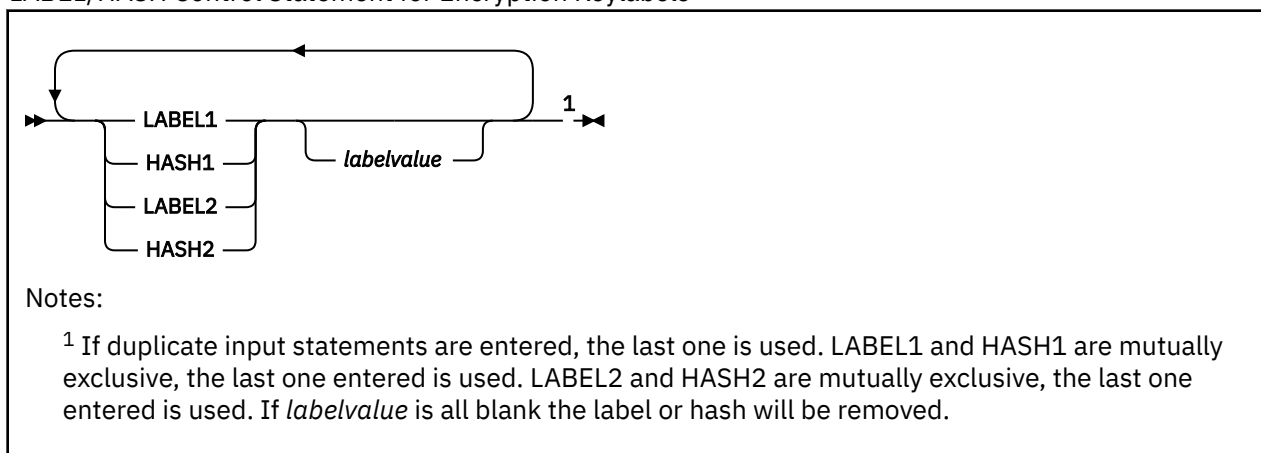
1. When the wrong input tape is mounted, the message HCP709E is displayed, and the tape rewinds and unloads, regardless of whether the options REWIND, UNLOAD, or LEAVE are specified.
2. If DDR is processed from CMS, failure to attach the tape drive or the disk device (or both) to your virtual machine before invoking the input/output statement causes the following response to be displayed:

```
HCP708E INVALID INPUT OR OUTPUT DEFINITION
```

3. The EMSG and ESKIP options are ignored when executing the stand-alone version of DDR in a real machine to dump or restore FBA data.
4. Options EMSG and ESKIP are mutually exclusive. The last option entered will be the one in effect.

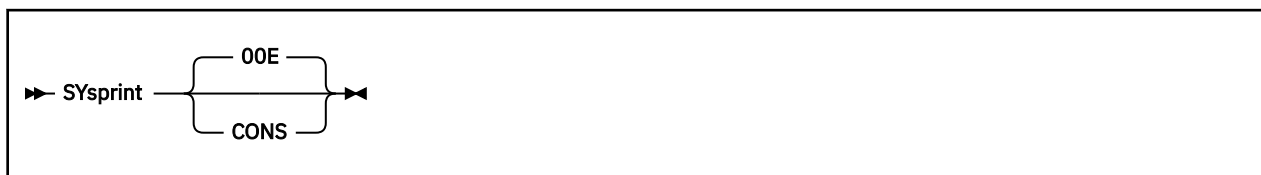
5. If LEAVE was specified and an End-Of-Tape condition is encountered while performing End-Of-Job processing, the tape will be rewound and unloaded and message HCP8102I will be issued.
6. Tape hardware compaction is the default on 3490 and later tape drives. Hardware compaction may give better compression than software compaction. On 3480 tape drives, if both the dumping and restoring tape drive have XF capability, then MODE XF may be used to turn on tape hardware compaction. Compact or LZcompact may be used for 3480 tape drives that do not have XF support and will automatically turn off tape hardware compaction on any tape drive.
7. When the KEY option is used for tape encryption, the tape must be at the beginning. If encrypted data has already been written to the tape and the tape is not at the beginning, any further data dumped to the tape without using the KEY option will also be encrypted.

#### LABEL/HASH Control Statement for Encryption Keylabels



Up to two key labels may be specified for tape encryption. LABEL1/LABEL2 indicates the label value should be used as is. HASH1/HASH2 indicates the key label encoding mechanism is a hashed value of the associated public key, to better share data accessed through different key managers. Label values are up to 64 characters. The first character of the key label must be alphabetic or a national character (#, \$, @). The remaining characters may be alphabetic, numeric, national, or a period. Imbedded blanks are allowed.

#### SYSPRINT Control Statement



The SYSPRINT control statement describes the device to which output is to be sent. If you specify the SYSPRINT CONS option, the output is directed to the console.

In the CMS environment, all output is directed (by default) to CMS printer 00E, unless you specify the SYSPRINT CONS option. Specifying SYSPRINT CONS will direct all output to the console. Any other SYSPRINT option specification is ignored.

#### 00E

the device number of the printer to be used for printed output. This is the default.

#### CONS

the console as the output device.

**Note:** If the SYSPRINT control statement is issued with anything other than 00E or CONS and the option is not omitted, the statement is ignored and you will receive the message:

```
HCP701I SYSPRINT NOT 00E OR CONS--IGNORED
```

#### PROMPTS Processing Control Statement

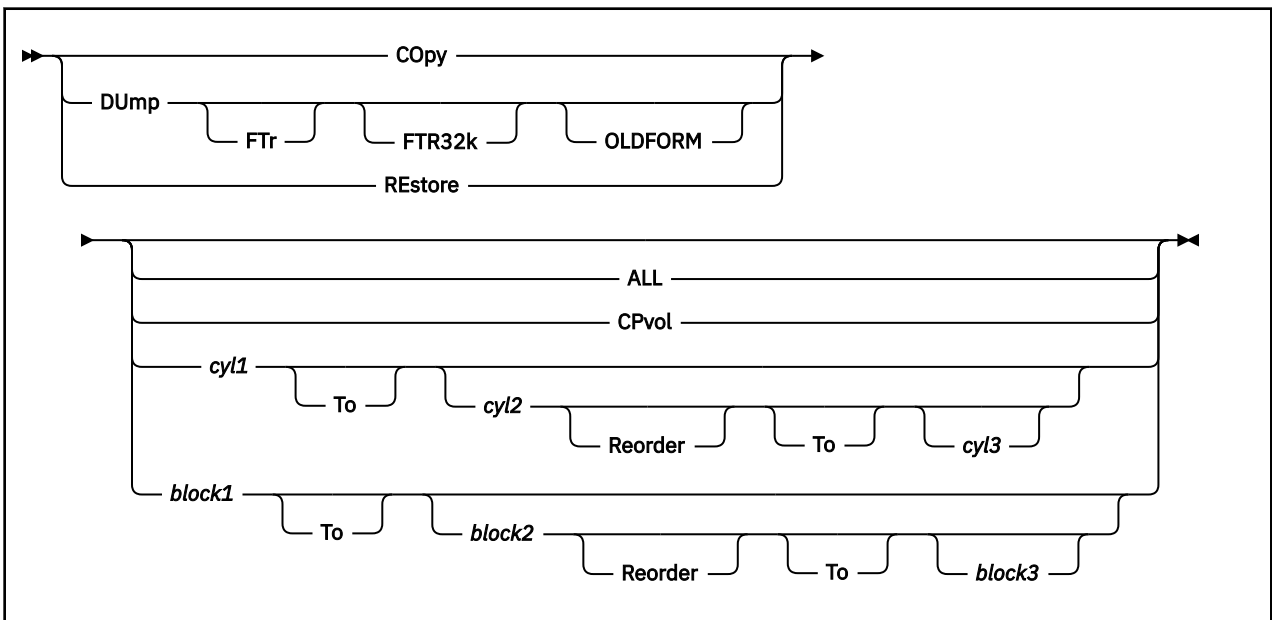


The PROMPTS processing control statement provides a user with the option to bypass the prompts that are normally issued with messages 711, 716, 717, and 725. PROMPTS ON (the default) will cause DDR to solicit a response to any of the above mentioned messages from the user before any further processing will be done. The action that DDR will take depends on the response entered by the user. PROMPTS OFF will cause DDR to act as if every prompt encountered has been answered with "YES". Although the prompting messages will no longer be issued when PROMPTS OFF is in effect, other messages similar in content to the original messages will be issued. These other messages, numbered 696-699, are the PROMPTS OFF equivalents of messages 711, 716, 717, and 725, respectively.



**Attention:** Although the default is to prompt the user with messages 711, 716, 717, and 725 when unexpected circumstances arise, these prompts when turned off will remain off for the rest of that session unless the user turns them back on by entering PROMPTS ON. DDR users considering the use of PROMPTS OFF should carefully examine the conditions under which the prompting messages are issued and then determine whether any adverse effects would result by allowing the DDR function to continue.

Function Statements



The function statements tell the DDR utility the action to perform. The function commands also describe the extents to be dumped, copied, or restored.

**Note:** The FTR, FTR32k, and OLDFORM options are valid only with the DUMP control statement.

**DUmp**

requests that the utility move data from a direct access volume onto a magnetic tape or tapes. After the data is dumped, if the COMPACT or LZCOMPACT option was specified on the OUTPUT control statement, you receive the following messages:

```
'BYTES IN _____ BYTES OUT _____'
'TRACKS NOT COMPACTED ON THE TAPE - _____'
```

or, for an FBA DASD:

```
'BYTES IN _____ BYTES OUT _____'
'BLOCKS NOT COMPACTED ON THE TAPE - _____'
```

where

- “BYTES IN” is the total number of bytes that were read in from the input device.
- “BYTES OUT” is the total number of bytes that written out to the output (tape) device.
- “TRACKS or BLOCKS NOT COMPACTED ON THE TAPE” is the number of Tracks or Blocks read in from DASD that were not reduced in size by software compaction, and therefore were written out exactly as they were read in.

If you know the density of the output tape, you can use the value for “BYTES OUT” to calculate the approximate number of feet of tape used. The density is dependent on the tape device and may be modified with the value for MODE on the OUTPUT control statement.

The format of the tape depends on the type of the direct access volume.

Once the dump is complete, DDR writes a tape mark.

### **FTr**

requests an output tape format of variable unblocked records with a maximum size of 65535. Specifying FTR will reduce the number of read I/O operations initiated on the device by reading many records (a full-track) for each operation. In addition, the tape record format (variable length and unblocked) will result in less tape used. The size of the records and the number of records on each track written to the tape depends on the density of the tape. The option can be used for those devices supporting the full-track-read feature (FTR): 3380, 3390 and for FBA devices. FTR is the default for the 3380 and 3390 and therefore need not be specified.

If FTR is specified on the DUMP control statement for a count-key-data (CKD) DASD but the control unit does not support the feature, a message is written, and the operation proceeds with data written, in the old format.

For a count-key-data DASD, the data is moved cylinder by cylinder. Any number of cylinders can be moved. The format of the resulting tape is:

- *Non-FTR and Non-Compact Format:*

*Record 1:* a volume header record, consisting of data describing the volumes.

*Record 2:* a track header record, consisting of a list of count fields to restore the track, and the number of data records written on tape. After the last count field, the record contains key and data records to fill the 4 KB buffer.

*Record 3:* track data records, consisting of key and data records packed into 4 KB blocks, with the last record truncated.

*Record 4:* either the end-of-volume (EOV) or end-of-job (EOJ) trailer label. The end-of-volume label contains the same information as the next volume header record, except that the ID field contains EOJ. The end-of-job trailer label contains the same information as record 1, except that the cylinder number field contains the disk address of the last record on tape and the ID field contains EOJ.

- *FTR or Compact Format:*

*Record 1 and Record 4* are the same as described for the non-FTR format.

*Record 2:* a track header record, consisting of fields containing the length of the track, the density of the tape, and the number of count fields in the track followed by the track contents.

*Record 3:* track data records, consisting of count-key-data records in 8 KB, 12 KB, or 48 KB blocks for 800, 1600, or 6250 BPI respectively. The last block in all cases is a short block.

For FBA devices, the data is moved in sets of blocks. Each set is arbitrarily called a track (there is no relation to device geometry in this terminology). Any number of blocks can be moved with one DUMP statement. The format of the resulting tape depends on whether the FTR option is used or not, and also on the recording format of the tape.

- *FBA Tape Format:*

*Record 1:* a volume header record, consisting of data describing the volume

*Record 2:* a track header record. This consists of data that describes the set of blocks that follow (such as block numbers and the number of tape records required to hold these FB-512 blocks).

Following the control data are the actual FB-512 blocks filling out the tape record. When possible, DDR will use a 480K data buffer to store the FBA data (959 FBA blocks maximum). If storage for this buffer cannot be obtained or if the OLDFORM option was specified, then a smaller, 64K buffer will be used (127 FBA blocks maximum).

*Record 3:* FB-512 data records. These contain the rest of the blocks making up the set. The record lengths of records 2 and 3 are 4K when FTR is not specified. When FTR is specified, these tape records are 8K, 12K, or 48K depending on the recording format of the tape.

*Record 4:* either the end-of-volume (EOV) or end-of-job (EOJ). The end-of-volume label contains the same information as the next volume header record, except that the ID field contains EOV. The end-of-job trailer label contains the same information as record 1, except that the ID field contains EOJ.

**Note:** When dumping with FTR to a 3490E or subsequent tape device, tape records may vary in length (depending on the size of each track header record) up to a maximum record length of 64K–1.

### **FTR32k**

requests an output tape format of variable unblocked records with a maximum size of 32654 bytes. Using this option instead of FTR will possibly result in more I/O to the output device and is generally recommended only when an output tape will be read by applications that cannot handle larger blocks. Generally FTR (which is the default for many devices) is a better choice. See FTR for a complete description of this option.

### **OLDFORM**

indicates that the FBA data should be dumped using the smaller buffer (64K) rather than the larger buffer (480K) to remain compatible with older versions of DDR. The large buffer was implemented to enhance the performance of dumping FBA data to a streaming tape drive in a virtual machine environment. Using the OLDFORM option in this situation may degrade performance. This option is ignored when dumping from a CKD/ECKD DASD.

### **COPY**

requests that the utility copy data from one device to another device of the same or equivalent type. Data may be recorded on a cylinder basis from input device to output device. A tape-to-tape copy can be accomplished only with data dumped by this program. A tape in compact format may be used as input. For a tape-to-tape copy, the output is in the same format (compact, LZcompact, or standard) as the input tape. The COMPACT or LZCOMPACT option on the OUTPUT control statement is not valid for the COPY function. If it is specified, a message stating COMPACT OPTION IGNORED FOR COPY OPERATIONS is displayed.

### **REstore**

requests that the utility return data that has been dumped by this program. Data can be restored only to a DASD volume of the same or equivalent device type from which it was dumped. It is possible to dump from a real disk and restore to a minidisk as long as the device types are the same. A tape in a compact format may be used as input. DDR checks to see if the input is in compact format, and expands the data back to standard format, if needed. You do not need to specify anything to the program about the tape format. After data is restored, you receive the following message:

```
'BYTES RESTORED _____'
```

### **Notes:**

1. When restoring DDR-compacted tapes which were created with a release of DDR prior to release 2.0, the number of bytes indicated by the 'BYTES RESTORED' response will be larger than the number of bytes originally indicated by the 'BYTES IN' response following the dump function. This discrepancy is because DDR now uses a larger track header record format. The current track header record is 4 bytes longer than that used in release prior to release 2.0, thus the 'BYTES RESTORED' value will be larger than the 'BYTES IN' value by 4 bytes per track header record. This does not compromise the integrity of the data restored.
2. DDR reads past any tape marks found between dumps during a restore.

**cyl1 To cyl2 Reorder To cyl3**

Only those cylinders specified are moved, starting with the first track of the first cylinder (*cyl1*), and ending with the last track of the second cylinder (*cyl2*). The REORDER operand causes the output to be reordered, that is, moved to different cylinders, starting at the specified cylinder (*cyl3*) or at the starting cylinder (*cyl1*) if *cyl3* is not specified. The REORDER operand must not be specified unless specified limits are defined for the operation; the starting and, if required, ending cylinders (*cyl1* and *cyl2*) must be specified. Note that if the input device cylinder extents exceed the number of cylinders specified on the output device, an error message results.

**block1 To block2 Reorder To block3**

Only the specified blocks are moved, starting with the first block (*block1*), up to and including the last block (*block2*). The REORDER operand causes the data to be moved to a different DASD location. The REORDER operand must not be specified unless limits are defined for the operation. If *block1* or *block2* exceeds the capacity of the input drive, an error message will result. Similarly, if the reorder block, *block3*, exceeds the capacity of the output device, an error message will be issued. When the output device is a tape, the device capacity is considered to be the same as the capacity of the input FBA DASD or of the FBA volume residing on the input tape.

**CPvol**

specifies that cylinder 0, or blocks 0-31 (for FBA devices) and all permanent, parameter, and active directory space is dumped or copied as indicated by the z/VM allocation record from the input device. In the case of a tape input function specifying CPVOL, all the data is restored or copied.

**ALL**

specifies that the operation is to be performed on the entire DASD volume (all cylinders or all blocks).

**Note:** The occurrence of message HCP705E (issued upon completion of the copy, restore, or dump operation) indicates that an attempt was made to copy, restore, or dump the contents of DASD locations beyond the extents of the designated minidisk.

**Restrictions**

- Each track must contain a valid home address, containing the real cylinder and track location.
- Record zero must not contain more than eight key and data characters.
- Flagged tracks are treated just as any other track for all 2314 and 2319 devices—that is, no attempt is made to substitute the alternate track data when a defective primary track is read. In addition, tracks are not inspected to determine whether they were previously flagged when written. Therefore, volumes containing flagged tracks should be restored to the same cylinders of the volume from which they were dumped. The message HCP715E occurs each time a defective track is dumped, copied, or restored, and the operation continues.
- A minidisk that has extended into the alternate track cylinder must be moved either to another area of the disk, or to another disk entirely. Use the REORDER option to copy or restore a minidisk to another area.
- The FTR option can be used with the COMPACT option; however, DDR ignores the FTR option. This happens because when data compaction is requested, DDR defaults to using full-track-read if possible, and the tape record formats for the default are the same as those generated by FTR.
- For DASDs other than the 3380 and 3390 that support the full track read (FTR) processing, the option must be specified. Otherwise, the tape is produced in the current DDR format of 4096 blocks.
- Tapes created by the DDR DUMP function in compact or LZcompact format cannot be used as input to earlier levels of DDR that do not support the same format.
- If performing a DDR COPY from a device with a cylinder-based allocation record to a device with an extent-based allocation record, or vice versa, and if you plan on attaching the output DASD to the system as a CP volume, the output DASD should be reallocated by issuing the ICKDSF CPVOLUME ALLOCATE command. If the output DASD contains a cylinder-based allocation map when it should contain an extent-based allocation map, ICKDSF will convert the cylinder-based map into an extent-based map during the reallocation. Likewise, if the output DASD contains an extent-based allocation map when it should have a cylinder-based allocation map, ICKDSF will convert the extent-based map into a cylinder-based map during the reallocation.

*Example 1*

```
input 191 3390 sysres
output 180 3490 181dump all
input 130 3390 mini01
dump 1 to 50 reorder 51
dump 60 70 101
```

This example dumps all data from the volume labeled SYSRES onto the tape mounted on unit 180. If the program runs out of space on the first tape, it continues dumping onto the alternate device (181). A map of the dumped cylinders is printed on unit 00E while the program is dumping. When the first function is complete, the volume labeled MINI01 is dumped onto a new tape. Its cylinder header records are labeled 51 to 100. A map of the dumped cylinders is printed on unit 00E. Next, cylinders 60 to 70 are dumped and labeled 101 to 111. This extent is added to the cylinder map on unit 00E. When the DDR processing is complete, the tapes are unloaded, and the program stops.

*Example 2*

```
input 182 3490
output 250 fb-512 systst
restore 10 to 20 reorder 30
```

This example restores data previously dumped from an FBA device to another FBA device with device number 250. Blocks 10 through 20 stored on the tape are restored to the DASD on blocks 30 through 40.

If cylinder or block extents are being defined from the console, the user need only enter DUMP, COPY or RESTORE on the command line. The following is displayed:

```
ENTER CYLINDER EXTENTS
ENTER:
```

or

```
ENTER BLOCK EXTENTS
ENTER:
```

For any extent after the first extent, the following message is displayed.

```
ENTER NEXT EXTENT OR NULL LINE
ENTER:
```

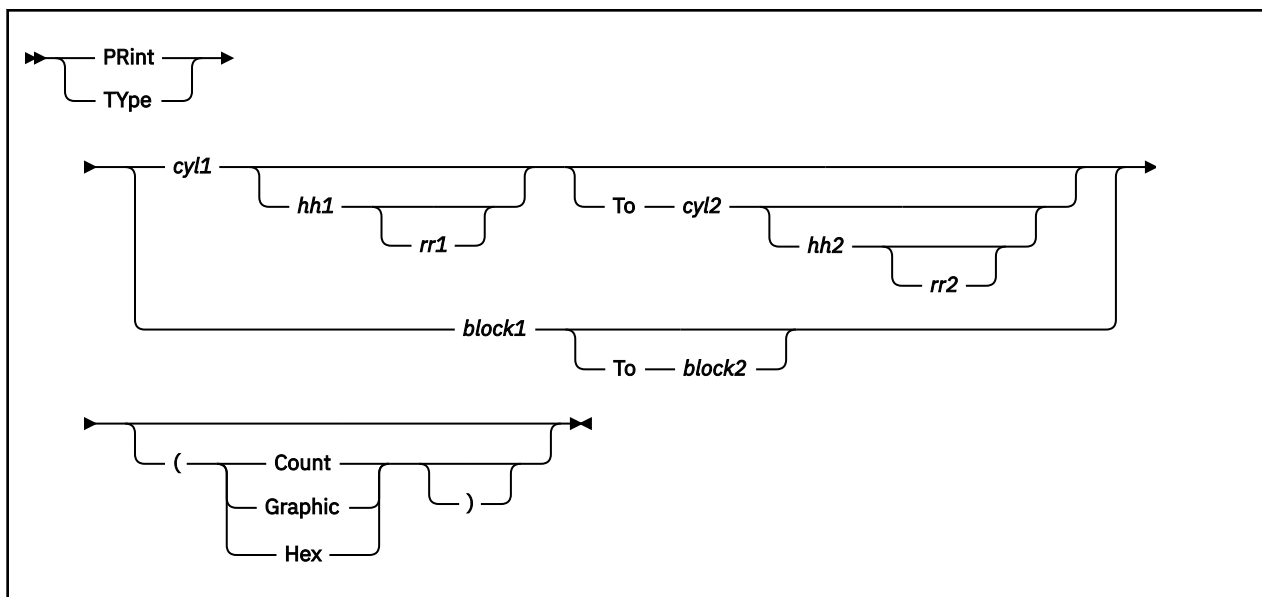
You may then enter additional extents to be dumped, restored, or copied. A null line causes the job step to start.

**Notes:**

1. When a cylinder map is printed on the virtual printer (00F as in the previous example) a heading precedes the map information. DDR displays the disk, time and zone in the heading.
2. Attempts to restore cylinders or blocks beyond the capacity that were recorded on the tape produces a successful EOJ, but the printout only indicates the last cylinder or block found on the tape.

PRINT/TYPE Function Statement





Use the PRINT and TYPE function statement to print or type (display) a hexadecimal and EBCDIC translation of each record or block specified. The output is directed to the system console for the TYPE function, or to the SYSPRINT device for the PRINT function. (This does not cause redefinition of the output unit definition.)

A tape in a compact format may be used as input to the PRINT or TYPE function. The printed or displayed output is not affected by the input format. You do not need to specify anything to the program about the format of the input tape.

#### **cyl1**

the starting cylinder. Up to an 8-digit cylinder value is accepted.

#### **hh1**

the starting track. If present, it must follow *cyl1*. The default is track zero.

#### **rr1**

the starting record. If present, it must follow *hh1*. The default is home address and record zero.

#### **TO cyl2**

the ending cylinder. If more than one cylinder is to be printed or typed, TO *cyl2* must be specified.

#### **hh2**

the ending track. If present, it must follow *cyl2*. The default is the last track on the ending cylinder.

#### **rr2**

the last record to print. The default is the last record on the ending track.

#### **block1**

identifies the first FBA device block that is to be printed or typed.

#### **To block2**

identifies the last FBA device block that is to be printed or typed. If more than one block is to be printed or typed, To *block2* must be specified.

#### **Hex**

prints or displays a hexadecimal representation of each record specified.

#### **Graphic**

prints or displays an EBCDIC translation of each record specified.

#### **Count**

prints or displays only the count field for each record specified. The COUNT option is ignored for FBA data.

#### *Usage Notes*

1. If the TYPE statement follows the occurrence of error message HCP705E and specifies the same cylinder, track, and record, or block extents indicated in the error message, the contents of the printed record or block must be interpreted in the context of the I/O error information given in the initial message.

**Examples**

```
print 0 to 3
```

prints all of the records from cylinders 0, 1, 2, and 3.

```
print  
0 1 3
```

prints only one record, from cylinder 0, track 1, record 3.

```
print  
1 10 3 to 1 15 4
```

prints all records starting with cylinder 1, track 10, record 3, and ending with cylinder 1, track 15, record 4.

The example in [Figure 1 on page 2163](#) shows the information displayed at the console (TYPE function) or system printer (PRINT function) by the DDR utility. The listing is annotated to describe some of the data fields.

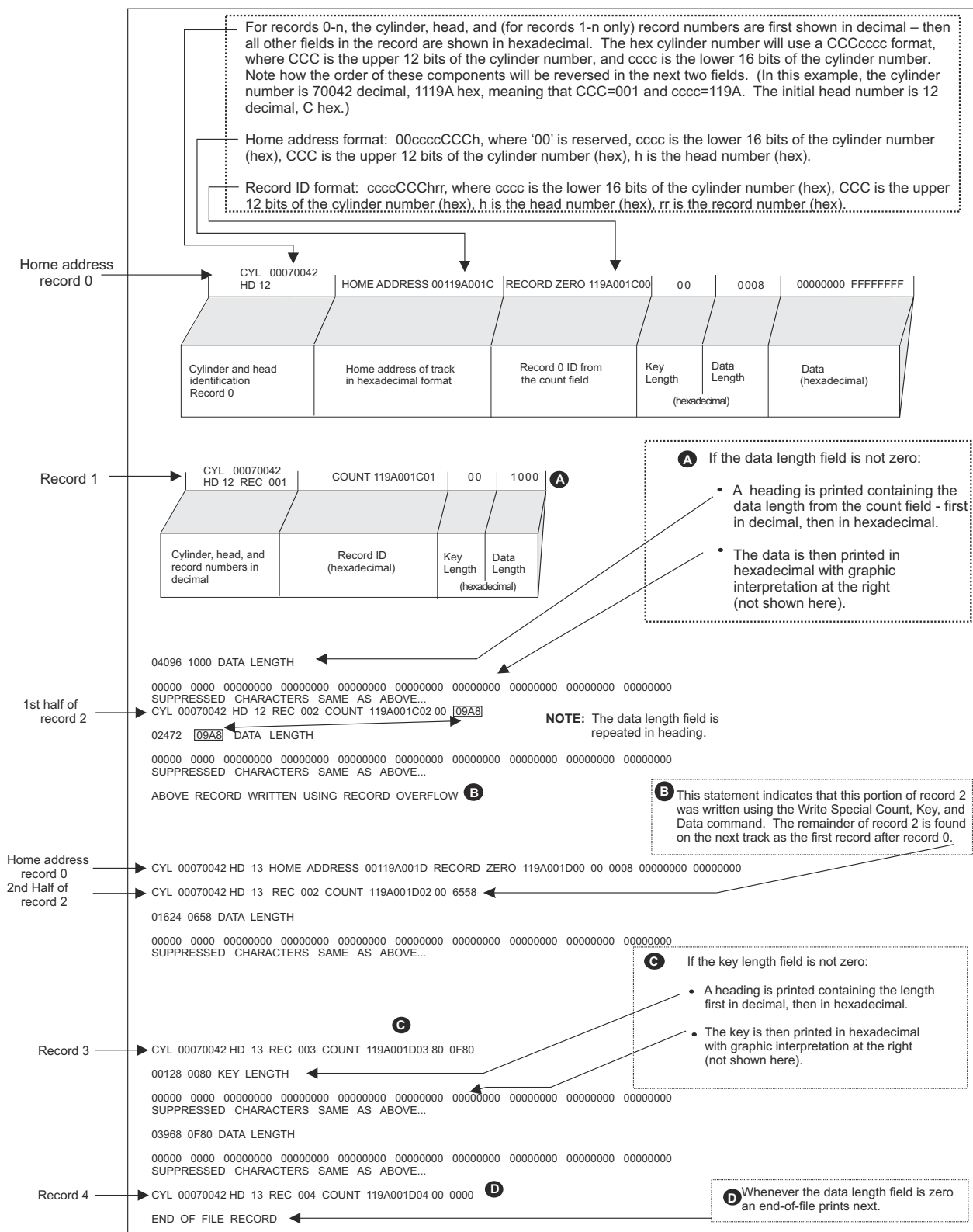


Figure 1. An Annotated Sample of Output from the TYPE and PRINT Functions of the DDR Program

For FBA device data, the utility first prints or types a heading that indicates the block number and then prints or types the 512 bytes of data residing in that block. This output will look as follows:

```

ENTER:
type 1
BLOCK 00000001
  
```

```

000000 0000 C3D4E2F1 D7D9D6D7 C5D90000 (Con't)
00000200 00000004 000002E8 000002E8 (Con't)
000002E8 *CMS1PROPER.....Y...Y...Y*
00032 0020 00000005 00000040 00000008 (Con't)
91031317 26230000 00000000 00000001 (Con't)
00000001 *.....j.....*
00064 0040 00000001 00000000 00000000 (Con't)
00000000 00000000 00000000 00000000 (Con't)
00000000 *.....*
00096 0060 00000000 00000000 00000000 (Con't)
00000000 00000000 00000000 00000000 (Con't)
00000000 *.....*
SUPPRESSED CHARACTERS SAME AS ABOVE ...

```

## Responses

**Note:** If PROMPTS OFF was specified by the user, messages 696-699 will replace messages 711, 716, 717, and 725, respectively. Please see the PROMPTS control statement description for more details. See also the list of DDR-issued messages at the end of this utility description.

```

VOLID READ IS valid2 [NOT valid1]
DO YOU WISH TO CONTINUE? RESPOND YES NO OR REREAD:

```

### **valid2**

the volume identifier from the VOL1 label on the DASD unit.

### **valid1**

the volume identifier from the INPUT or OUTPUT control card.

The volume identifier read from the device at *devno* is not the same as that specified on the INPUT or OUTPUT control card.

```

NO VOL1 LABEL FOUND FOR valid
DO YOU WISH TO CONTINUE? RESPOND YES NO OR REREAD:

```

### **valid**

the volume identifier of the DASD device from the INPUT or the OUTPUT control card.

The DASD at *devno* contains no volume identifier.

```

DATA DUMPED FROM valid1 TO BE RESTORED TO valid2
DO YOU WISH TO CONTINUE? RESPOND YES NO OR REREAD:

```

### **valid1**

the volume identifier from the input tape header record (volume dumped).

### **valid2**

the volume identifier from the output DASD.

This message is printed to verify the input parameters.

```

ENTER CYLINDER EXTENTS
ENTER:

```

or

```

ENTER BLOCK EXTENTS
ENTER:

```

This message is received only if you are entering input from your terminal.

```

END OF VOLUME CYL xxxxxxxx HD xx, MOUNT NEXT TAPE

```

or

```

END OF VOLUME BLOCK xxxxxx, MOUNT NEXT TAPE

```

DDR continues processing after the next tape is mounted.

```
RESTORING valid
```

**valid**

the volume identifier of the disk dumped.

The RESTORE operation has begun.

```
COPYING valid
```

The COPY operation has begun.

```
DUMPING valid
```

The DUMP operation has begun.

```
PRINTING valid
```

The PRINT operation has begun.

```
END OF DUMP
```

The DUMP operation has ended.

```
END OF RESTORE
```

The RESTORE operation has ended.

```
END OF COPY
```

The COPY operation has ended.

```
END OF PRINT
```

The PRINT operation has ended.

```
END OF JOB
```

All specified operations have completed.

```
ENTER:
```

Prompts for input from the terminal. A null line—that is, pressing the Enter key or equivalent—returns control to CMS.

```
ORIGINAL INPUT DEVICE WAS(IS) LARGER THAN OUTPUT  
DEVICE. DO YOU WISH TO CONTINUE?  
RESPONSE YES OR NO:
```

**Explanation:** RESTORE function—The number of cylinders or blocks on the original DASD input unit is compared with the number on the output device.

COPY function—The input device contains more cylinders or blocks than the output device.

**Note:** This response is issued only if the ALL operand was specified on the RESTORE or COPY function statement.

**Operator Action:** The operator must determine if the COPY or RESTORE function is to continue. The response is either yes or no.

A progress response will be issued to the user's console if the enter key is pressed while performing a dump, copy, or restore of data using the stand-alone version of the DDR utility (IPL DDRXA). The progress

response indicates the last block or the last cylinder and head successfully transferred to the output device at the time when the enter key was pressed.

The FBA format of the progress response is:

```
LAST BLOCK - xxxxxxxx
```

The CKD format of the progress response is:

```
LAST ADDRESS - CYL xxxxxxxx HD xx
```

## Messages

- HCP696I VOLID READ IS *valid1* NOT *valid2*
- HCP697I NO VOL1 LABEL FOUND FOR *valid1*
- HCP698I DATA DUMPED FROM *valid1* TO BE RESTORED TO *valid2*
- HCP699I SOURCE DASD DEVICE WAS (IS) LARGER THAN OUTPUT DEVICE
- HCP700E INPUT UNIT IS NOT A CPVOL
- HCP701E INVALID OPERAND - *operand*
- HCP701I SYSPRINT NOT 00E OR CONS--IGNORED
- HCP702E CONTROL STATEMENT SEQUENCE ERROR
- HCP703E OPERAND MISSING
- HCP704E DEV *rdev/vdev* NOT OPERATIONAL
- HCP705E I/O ERROR *rdev* IRB *irb* SNS *sense* CCW *ccw* {INPUT *bbcchh* OUTPUT *bbcchh* | INPUT *block* OUTPUT *block*} [DO YOU WISH TO CONTINUE? RESPOND YES OR NO:]
- HCP708E INVALID INPUT OR OUTPUT DEFINITION
- HCP709E WRONG INPUT TAPE MOUNTED
- HCP710A DEVICE *rdev* INTERVENTION REQUIRED
- HCP711D VOLID READ IS *valid* [NOT *valid2*] DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
- HCP712E NUMBER OF EXTENTS EXCEEDS 300
- HCP713E OVERLAPPING OR INVALID EXTENTS
- HCP714E {RECORD *bbcchh* | BLOCK *block* } NOT FOUND ON INPUT TAPE [DO YOU WISH TO CONTINUE? RESPOND YES OR NO:]
- HCP715E LOCATION {*bbcchh*} IS A FLAGGED TRACK
- HCP716D NO VOL1 LABEL FOUND [FOR *valid*] DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
- HCP717D DATA DUMPED FROM *valid1* TO BE RESTORED TO *valid2*. DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
- HCP718E OUTPUT UNIT IS {FILE PROTECTED – TAPE|WRITE INHIBITED – DASD} {*rdev/vdev*}
- HCP719E INVALID FILENAME OR FILE NOT FOUND
- HCP720E ERROR IN *routine*
- HCP721E RECORD *cchhr* NOT FOUND
- HCP725D SOURCE DASD DEVICE WAS (IS) LARGER THAN OUTPUT DEVICE. DO YOU WISH TO CONTINUE? RESPOND YES OR NO:
- HCP726E MOVING DATA INTO ALTERNATE TRACK CYLINDER(S) IS PROHIBITED
- HCP728E DECODE ERROR ENCOUNTERED: *nn*
- HCP729I FULL TRACK READ NOT AVAILABLE
- HCP731I COMPACT OPTION IGNORED FOR COPY OPERATIONS

- HCP756E PROGRAM CHECK PSW=*psw*. CODE=*code*
- HCP814E INSUFFICIENT STORAGE AVAILABLE FOR RESTORE
- HCP2236E TAPE *rdev* THE MAXIMUM BLOCK HAS BEEN EXCEEDED
- HCP2238E *rdev devname* A CHANNEL INTERFACE ERROR HAS OCCURRED
- HCP2239E TAPE *rdev* AN OPERATION WAS TERMINATED BECAUSE THE END OF DATA WAS ENCOUNTERED
- HCP2240E *rdev devname* A CHANNEL PROTOCOL ERROR HAS OCCURRED
- HCP2253E DEVICE *rdev* OPERATION TERMINATED DUE TO KEY ENCRYPTION FAILURE
- HCP2255E DEVICE *rdev* DOES NOT SUPPORT HARDWARE ENCRYPTION
- HCP2256E UNEXPECTED EOF WHILE READING/WRITING TO PIPE
- HCP6370E TAPE *rdev* MEDIUM IS NOT FORMATTED
- HCP8100I DEFAULT PRINTER 000E DOES NOT EXIST
- HCP8101E VERIFY PATH *yy* TO DEVICE *xxxx*; A POTENTIAL CONFIGURATION CHANGE WAS DETECTED.
- HCP8102I END OF VOLUME ENCOUNTERED ON TAPE *rdev*
- HCP9014W MACHINE CHECK
- HCP9037W {SAPL|DDR} requires z/Architecture to be installed.
- HCP9044W UNEXPECTED CHANNEL ERROR

The DDR utility issues the following return codes:

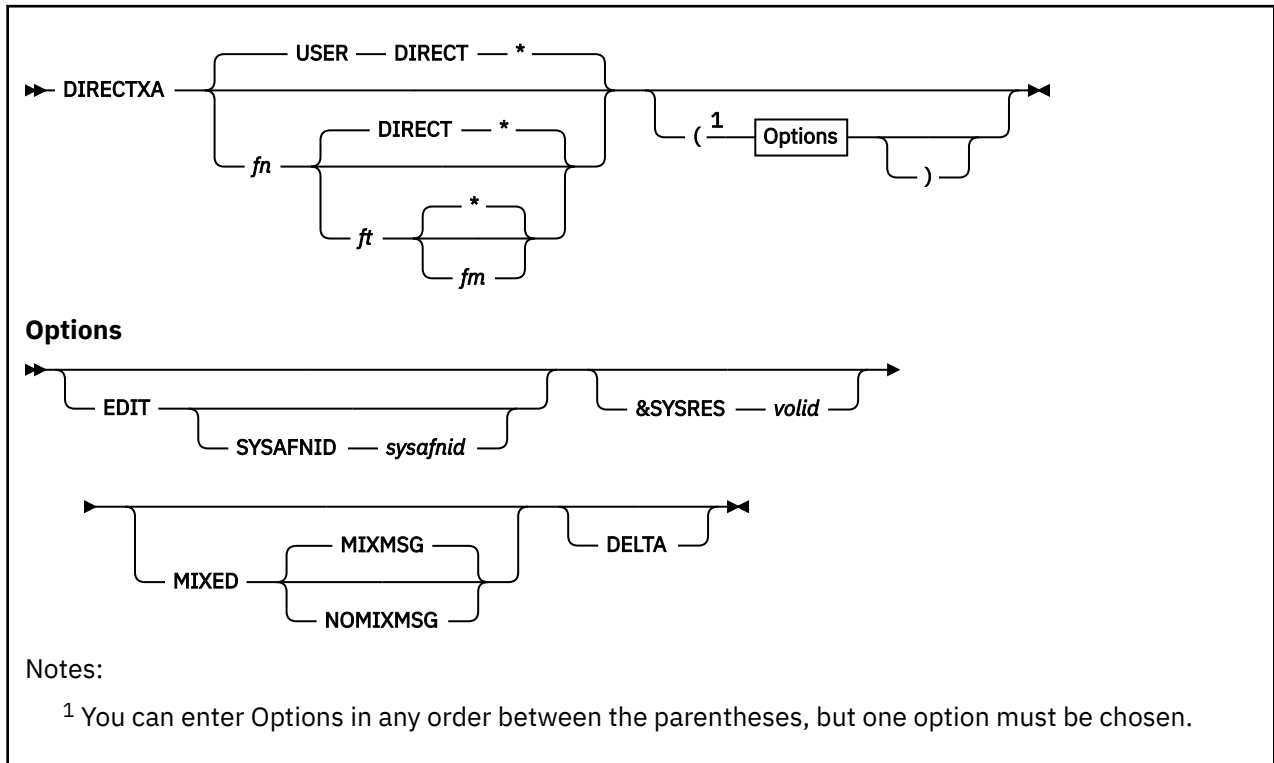
#### Code

#### Meaning

- 1**  
Invalid file name or file not found
- 2**  
Error running the program
- 3**  
Flagged DASD track
- 4**  
Permanent tape or DASD I/O error
- 1xx**  
Error in printing a line (PRINTL routine)
- 2xx**  
Error in reading from the terminal (LINERD routine)
- 3xx**  
Error in reading a file (FSREAD routine)
- 4xx**  
Error in writing to the terminal (LINEWRT routine)
- 20**  
Error in the decoding routine.

**Note:** xx is the return code from the CMS routine.

## DIRECTXA



### Purpose

Use the DIRECTXA utility to create a user directory.

### Operands

*fn*

identifies the file name of the source directory. If it is not specified, the default file name is USER.

*ft*

identifies the file type of the source directory. If it is not specified, the default file type is DIRECT.

*fm*

\*

identifies the file mode of the source directory. If it is not specified, the default file mode is asterisk (\*). (The utility searches all accessed disks for the file name and file type).

### Options

#### EDIT

lets you enter the DIRECTXA command without updating the directory on disk. With this option, you can check the syntax of directory statements.

#### SYSAFNID *sysafnid*

identifies a 1- to 8-character alphanumeric value that must correspond to a system affinity ID specified on one of the DIRECTORY statements in the source directory named in the DIRECTXA command. A maximum of 56 system affinity ids can be specified. This option can be used only with the EDIT option, and it specifies that the DIRECTXA utility is to compile the source directory as if it were executing on the system identified by *sysafnid*.



This option is not allowed when processing an SSI-ready or SSI-enabled source directory. For more information about SSI-ready and SSI-enabled source directories, see [Source Directory Types](#) in *z/VM: CP Planning and Administration*.

### **&SYSRES valid**

is a one to six-character value that is used to represent the volume serial number of the system residence volume for minidisks defined using the symbolic value &SYSRES. This value is used in responses to the VMUDQ LSTMDISK function. It need not be the actual system residence volume serial number. If this option is omitted, a value of +VMRES is used.

If SSI is specified on the DIRECTORY directory statement in the source directory, then the &SYSRES option is not allowed, and &SYSRES and +VMRES are not allowed on MDISK directory statements. If this rule is violated for an SSI-enabled source directory, then an error message occurs and the object directory is not written. For an SSI-ready source directory, a warning message is issued explaining that &SYSRES or +VMRES are not allowed once the SSI option is added.

For more information about SSI-ready and SSI-enabled source directories, see [Source Directory Types](#) in *z/VM: CP Planning and Administration*.

### **MIXED**

specifies that DIRECTXA is to validate, but not process, statements and options encountered in the source directory that are unique to other releases of VM, such as VM/XA SP™ 2.1, VM/SP HPO release 5, and VM/ESA. This lets you use a source directory from one of those systems without having to edit out those unique values. If the MIXMSG option is in effect, DIRECTXA prints informational messages for each statement or option not processed.

### **MIXMSG**

#### **NOMIXMSG**

specifies whether to enter a message indicating that a directory statement or an option of a directory statement applies to other releases of VM, such as VM/XA SP 2.1, VM/SP HPO release 5, and VM/ESA. The default is MIXMSG, which causes informational messages to be issued for each statement or option not processed; see the description of MIXED above.

### **DELTA**

specifies that a mini-directory representing a set of changes from the existing base directory is to be created. This option causes the following:

- No attempt is made to bring the new directory online to CP
- The volume allocation map is not updated
- The volume label is not updated
- A password of DELETEUSER is allowed on the USER statement and marks the associated user deleted in the directory userid index.

The DELTA option is not allowed if the source directory contains IDENTITY or SUBCONFIG statements.

### How the Directory Program Works

The DIRECTXA utility performs the following VM steps during execution:

1. The program looks for the file you specify on the DIRECTXA command line. If you do not specify a file name or a file type, the program looks for a file that has the file name of USER and a file type of DIRECT.

The source directory found must be compatible with the system on which DIRECTXA is running (a system with no SSI cluster defined, a single-member SSI cluster, or a multiple-member SSI cluster), else the object directory is not brought online. For more information, see [Source Directory Types](#) in *z/VM: CP Planning and Administration*.

2. If the program does not find the source directory or an error occurs during processing, the program does not create a directory, and the old directory remains online. The program continues to check the syntax of all directory statements before ending.
3. DIRECTXA looks for the RPWLIST DATA file, which contains a list of restricted passwords. The RPWLIST DATA file is supplied on your system. You can edit this file to add your own restricted

passwords. If the RPWLIST DATA file is not found, DIRECTXA issues a warning message but continues processing. If the RPWLIST DATA file is found, DIRECTXA checks all the passwords in the directory against it and changes any restricted passwords (except the password of the user issuing the DIRECTXA command) to NOLOG. Any user whose password is changed to NOLOG cannot log on. If the password of the virtual machine issuing the DIRECTXA command is restricted, it is not changed to NOLOG. DIRECTXA issues an error message and does not update the directory.

4. If DIRECTXA runs out of DASD space, the utility issues an error message, and the directory is not processed. To continue, you must allocate more cylinders or blocks for the user directory and reissue the DIRECTXA command.

If DIRECTXA runs out of virtual storage, the utility issues an error message, and the directory is not processed. To continue, you must define more virtual storage (DEFINE STORAGE command), re-IPL CMS, and reissue the DIRECTXA command. Using the following formula, calculate the size of virtual storage you need:

```
vstor = 200 KB + (600 bytes * (users + totgrents))
          + (110 bytes * (device statements))
          + (250 bytes * (NAMESAVE + CPU statements))
          + (125 bytes * (IUCV + APPC statements))
          + length of SCPDATA specified on LOADDEV statements
          + length of commands specified on COMMAND statements
```

This is an estimate for the size of the directory. It is not an exact calculation.

**vstor**

the amount of virtual storage in the program area that DIRECTXA requires.

**userid**

identifies the total number of virtual machines and subconfigs defined in the directory control file. This is the sum of all USER, IDENTITY, and SUBCONFIG statements plus the number of users defined by POOL statements.

**totgrents**

identifies the total number of POSIXGROUP statements, plus all groups listed in all POSIXGLIST statements, plus all groups specified in all POSIXINFO statements, plus 1.

5. If no errors occur, and EDIT is not specified, the program writes the directory on DASD and the DASD address of the new directory on the volume label for that device. If the program is updating an active system directory, the program makes the new directory immediately available for the system.

**Usage Notes**

1. On a system where different release levels of z/VM can be IPLed, DIRECTXA is required to be at the highest release level of z/VM that can run on the system. When migrating to a higher release level of z/VM, run the highest release level of DIRECTXA to update the object directory prior to IPLing the higher release level of z/VM. An object directory created using the highest release level of DIRECTXA is compatible with all supported release levels of z/VM. Running a higher release level of z/VM with an object directory created using a lower release level of DIRECTXA is not supported.
2. By default, the DIRECTXA utility is located on the cross release utilities disk (PMAINT 551). This disk must be accessed to run the DIRECTXA utility.
3. You must have CP privilege class A, B, or C to update the system user directory.
4. A complete description of the directory program, including a description of directory statements, is contained in [Creating and Updating a User Directory in z/VM: CP Planning and Administration](#).
5. When writing a user directory, DIRECTXA does not overwrite the current directory, but does write an alternate user directory. The directory pointer in the volume label is then changed to refer to this new directory.
6. You should be careful to target a different physical pack when using the DIRECTXA utility from a new version of z/VM to create a user directory for that new version of the system, by running it under the old system.

7. If the directory specified on the DIRECTXA command is not the current directory, the directory that is created by this utility is not brought online.
8. The MIXED, MIXMSG, NOMIXMSG, EDIT and SYSAFNID options are not positional and can be entered in any order. However, if the SYSAFNID option is specified, the EDIT option also must be specified. If the MIXMSG or NOMIXMSG option is specified, the MIXED option also must be specified.

## Examples

To compile the USER DIRECT source directory and apply it to the system if successful, enter:

```
directxa user direct *
```

To do a test compile of a USER DIRECT source directory from a VM/SP HPO 5 system to see which control statements are HPO-only or contain HPO-only options, enter:

```
directxa user direct * (edit mixed
```

To do a test compile of a USER DIRECT source directory for a specific system named SYSVMA in a mixed complex of systems sharing the source directory, enter:

```
directxa user direct * (edit mixed mixmsg sysafnid sysvma
```

To do a test compile of a USER DIRECT source directory for a specific system named SYSVMA in a z/VM-only complex of systems sharing the source directory, enter:

```
directxa user direct * (edit sysafnid sysvma
```

To compile a USER DIRECT source directory for this system and to suppress the compatibility error messages, enter:

```
directxa user direct * (mixed nomixmsg
```

## Responses

Response 1:

When you enter the DIRECTXA command, you receive the following response:

```
z/VM USER DIRECTORY CREATION PROGRAM - VERSION n RELEASE n.n
```

You then receive Response 2, 3, or 4.

Response 2:

If the directory is updated and CP places the new directory in use, you receive the following response:

```
EOJ DIRECTORY UPDATED AND ON LINE
```

Response 3:

If the directory is written to a CP-formatted DASD that is not the CP-owned volume containing the online object directory, you receive the following response:

```
EOJ DIRECTORY UPDATED
```

Response 4:

You receive the following response if one or more of the following is true:

- You issued DIRECTXA with the EDIT option.
- Your virtual machine does not have the necessary privilege class to change the directory that is in use.
- You attempted to write a directory to the online object directory volume that is not valid in the current system configuration.

- Errors prevented the directory from being updated.

E0J DIRECTORY NOT UPDATED

## Messages

- HCP480E AN OPERAND OF THE LOAD STATEMENT IS MISSING.
- HCP481E THE LOAD STATEMENT TYPE FIELD IS INVALID; ONLY “C”, “G”, “P”, “I”, “S”, OR “U” IS ALLOWED.
- HCP482E THE FORMAT OF AN OPERAND ON THE LOAD STATEMENT IS INVALID.
- HCP483E THE CMS STATE COMMAND FAILED WITH RETURN CODE *nnnn* WHILE TRYING TO LOCATE THE DEFINITION FILE SPECIFIED ON THE LOAD STATEMENT.
- HCP485I THE DIRECTORY PROCESS IS RUNNING IN CPUID = *nnnnnn-xxxx*, WITH SYSTEM AFFINITY NAME = *aaaaaaaa*.
- HCP486E THE DIRECTORY CONTROL STATEMENT HAS AN INVALID CPUID FIELD.
- HCP487E NO DIRECTORY CONTROL STATEMENT WAS FOUND THAT MATCHES OUR CPUID (*nnnnnn-xxxx*).
- HCP489E THIS DIRECTORY CONTROL STATEMENT DOES NOT HAVE A SYSTEM AFFINITY ID.
- HCP490E THE EDIT MODE DIRECTORY STATEMENT MUST BE LAST IN THE SEQUENCE OF THE DIRECTORY STATEMENTS.
- HCP491E A USER STATEMENT DOES NOT DIRECTLY FOLLOW A SYSTEM AFFINITY PREFIX STATEMENT AT (RECORD NUMBER *nnnnnnnn*) IN FILE *filename filetype filemode*.
- HCP492I LOAD COMPLETED, STATEMENTS READ FROM CLUSTER FILES = *nnnnnnnn*, CLUSTER FILE DEFINITIONS = *nnnnnnnn*, SEPARATE FILE DEFINITIONS = *nnnnnnnn*, TOTAL STATEMENTS PROCESSED = *nnnnnnnn*.
- HCP493E THE OPTION *option* WAS SPECIFIED ON THE DIRECTXA COMMAND WITHOUT A VALUE FOLLOWING IT.
- HCP494I DIRECTORY OCCUPIES *nnnnn* DISK PAGES.
- HCP496E SSI option on the DIRECTORY statement conflicts with system affinity specifications.
- HCP497E An IDENTITY, SUBCONFIG or BUILD statement conflicts with system affinity specifications
- HCP498E BUILD statements are only allowed within IDENTITY definitions.
- HCP499E The number of BUILD statements exceeds the limit following IDENTITY userid.
- HCP642E INVALID POOL VALUE - *value*
- HCP664E System *system-id* was specified on multiple BUILD statements following IDENTITY *userid*.
- HCP664E SUBCONFIG ID *subconfig-id* was specified on multiple BUILD statements.
- HCP665E A SUBCONFIG definition does not exist for *id*
- HCP666E SUBCONFIG *id* is not specified on any BUILD statements
- HCP667E The object directory on volume (*volser*) is not enabled for use in a multi-member SSI cluster and will not be brought online.
- HCP668E The SSI-enabled object directory on volume (*volser*) is not valid for this configuration and will not be brought online.
- HCP669I [&SYSRES | +VMRES] WILL NOT BE ALLOWED ON THE [DIRECTXA COMMAND | MDISK STATEMENT] IF THE SSI OPTION IS ADDED TO THE DIRECTORY STATEMENT.
- HCP670E [&SYSRES | +VMRES] IS NOT ALLOWED ON THE [DIRECTXA COMMAND | MDISK STATEMENT] WHEN THE SSI OPTION IS SPECIFIED ON THE DIRECTORY STATEMENT.
- HCP671E If SSI is specified on the DIRECTORY statement, the system identifier on the BUILD statement cannot be \*.

- HCP672E If SSI is not specified on the DIRECTORY statement, the system identifier on the BUILD statement must be \*.
- HCP675E The VDISK operand is not allowed on the MDISK statement following IDENTITY *userid*
- HCP676E IDENTITY/SUBCONFIG statements are not allowed if the DIRECTXA DELTA option is specified.
- HCP677E A mix of SSI and non-SSI DIRECTORY statements is not allowed.
- HCP678E Number of volids specified on the DIRECTORY statement(s) exceeds the limit.
- HCP750E RESTRICTED PASSWORD FILE[, ERROR DURING READ| HAS BAD RECORD FORMAT| HAS BAD RECORD LENGTH]
- HCP751E INVALID OPERAND {FOLLOWING PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id* | PRECEDING A POOL STATEMENT} - {COMMAND LENGTH EXCEEDED | MAXIMUM COMBINED COMMAND LENGTH EXCEEDED}
- HCP752E STATEMENT SEQUENCE ERROR [FOLLOWING PROFILE/USER/IDENTITY/SUBCONFIG {*profname|userid|id*}]
- HCP753E OPERAND MISSING [{FOLLOWING PROFILE *profname* | FOLLOWING USER *userid* | ON POOL STATEMENT | FOLLOWING IDENTITY *userid* | FOLLOWING SUBCONFIG *id*}]
- HCP754E CONFLICTING OPERAND - *operand* [FOLLOWING PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*]
- HCP755E INCORRECT *keyword* SPECIFICATION FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}
- HCP758E DUPLICATE {CONSOLE | UNIT} DEFINITION FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}
- HCP759E INSUFFICIENT VIRTUAL STORAGE
- HCP760E NOT ENOUGH SPACE ALLOCATED FOR DIRECTORY
- HCP761E VOLID READ IS *valid1* NOT *valid2* ON *vdev*
- HCP762E DIRECTORY STATEMENT MISSING
- HCP763E INVALID FILENAME OR FILE NOT FOUND
- HCP764E ERROR IN *rdbuf*
- HCP765E INVALID CLASS DEFINITION - *class*
- HCP766E DUPLICATE CLASS DEFINITION - *class*
- HCP767I PASSWORD CHANGED TO NOLOG FOR *userid*
- HCP771E RESTRICTED PASSWORD AND NOLOG INVALID FOR *userid*
- HCP772E THE {PROFILE *profname* | USERID *userid* | POSIXGROUP *groupname*} WAS PREVIOUSLY DEFINED.
- HCP773E MINIOPT STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid* | SUBCONFIG *id*} DOES NOT APPLY TO VIRTUAL DEVICE *vdev* BECAUSE IT IS A FULL PACK MINIDISK.
- HCP774E I/O ERROR *vdev* UNIT EXCEPTION
- HCP775I ALTERNATE DEVICE *xxxx* IN USE
- HCP776E DUPLICATE DIRECTORY CONTROL STATEMENT FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}
- HCP777E DASDOPT STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid* | SUBCONFIG *id*} DOES NOT APPLY TO VIRTUAL DEVICE *vdev* BECAUSE IT IS NOT A FULL PACK MINIDISK.
- HCP782I ERROR ATTEMPTING TO RETURN FREE STORAGE
- HCP782E ERROR ATTEMPTING TO GET FREE STORAGE
- HCP783I OPERAND IGNORED - *operand* [FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}]
- HCP784E I/O ERROR *vdev* DEVICE NOT ATTACHED

- HCP786I EDIT CHARACTER - *value* IS NOT VALID FOLLOWING {USER *userid* | IDENTITY *userid*}
- HCP787E I/O ERROR *vdev* INCORRECT RECORD LENGTH DETECTED
- HCP788E PERMANENT I/O ERROR *vdev*
- HCP789E NUMBER OF CPU STATEMENTS EXCEEDS CPU LIMIT FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid*}
- HCP790E CPU STATEMENT CONTAINS INVALID CPU ADDRESS FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid*}
- HCP791E DUPLICATE CPU DEFINITION FOLLOWING {PROFILE *profname* or USER *userid* | IDENTITY *userid*}
- HCP792E A BASE CPU HAS ALREADY BEEN DEFINED FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid*}
- HCP793E LOGICAL RECORD LENGTH OR RECORD FORMAT IS INCORRECT
- HCP795E STATEMENT CONFLICTS WITH {CLASS VALUE | STORAGE SIZE} ON {USER | IDENTITY} STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid*}
- HCP796E MAXIMUM NUMBER OF CRYPTOS ALREADY DEFINED ON CPU STATEMENTS FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid*}
- HCP797E THE VALUE OF *keyword1* CONFLICTS WITH THE VALUE OF *keyword2* FOLLOWING {IDENTITY *userid* | SUBCONFIG *id*}.
- HCP798E *keyword1* KEYWORD CONFLICTS WITH *keyword2* FOLLOWING {PROFILE *profname* or USER *userid* | IDENTIFY *userid* | SUBCONFIG *id*}
- HCP1754I DISK SPACE ON THE DIRECTORY VOLUME IS TOO FRAGMENTED-CANNOT CREATE NEW DIRECTORY EXTENT
- HCP1755E THERE IS NO POSIXGROUP DEFINED WITH {GID *gid* | GNAME *gname*} FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid*}.
- HCP1756E THE *statement* STATEMENT WAS CONTINUED PAST END OF FILE
- HCP1757E THE *statement* STATEMENT EXCEEDS ITS ALLOWED MAXIMUM LENGTH FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}
- HCP1758E DUPLICATE {GID *gid* | GNAME *gname*} SPECIFIED IN THE POSIXGLIST STATEMENT FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid*}
- HCP1759E *operand* HAS ALREADY BEEN SPECIFIED FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}.
- HCP1760E AN OPERAND EXCEEDS ITS ALLOWED MAXIMUM LENGTH FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}
- HCP1762E INVALID CYL/BLK SPECIFICATION CODE= *nn* FOLLOWING {USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}
- HCP1763E I/O ERROR *vdev* VIRTUAL DEVICE IS BUSY
- HCP1764E MSGPROC WAS SPECIFIED ON A SPECIAL STATEMENT BUT NEITHER CFUSER NOR CFVM WAS SPECIFIED FOLLOWING USER *userid*
- HCP1764E MSGPROC WAS SPECIFIED ON A SPECIAL STATEMENT BUT NEITHER CFUSER NOR CFVM WAS SPECIFIED FOLLOWING PROFILE *name*
- HCP1764E MSGPROC WAS SPECIFIED ON A SPECIAL STATEMENT BUT NEITHER CFUSER NOR CFVM WAS SPECIFIED FOLLOWING IDENTITY *userid*
- HCP1764E MSGPROC WAS SPECIFIED ON A SPECIAL STATEMENT BUT NEITHER CFUSER NOR CFVM WAS SPECIFIED FOLLOWING SUBCONFIG *id*
- HCP1764E MSGPROC WAS SPECIFIED ON A SPECIAL STATEMENT AND DEVICE COUNT IS INVALID OR OUT OF RANGE FOLLOWING USER *userid*
- HCP1764E MSGPROC WAS SPECIFIED ON A SPECIAL STATEMENT AND DEVICE COUNT IS INVALID OR OUT OF RANGE FOLLOWING PROFILE *name*

- HCP1764E MSGPROC WAS SPECIFIED ON A SPECIAL STATEMENT AND DEVICE COUNT IS INVALID OR OUT OF RANGE FOLLOWING IDENTITY *userid*
- HCP1765E THE PROFILE STATEMENT HAS NO NAME.
- HCP1766E THE PROFILE NAME ON THE PROFILE STATEMENT EXCEEDS 8 CHARACTERS.
- HCP1766E THE PROFILE NAME ON THE INCLUDE STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid*} EXCEEDS 8 CHARACTERS.
- HCP1766E THE PROFILE NAME ON THE INCLUDE STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid*} NOT FOUND.
- HCP1766E THE PROFILE NAME ON THE POOL STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid*} EXCEEDS 8 CHARACTERS.
- HCP1766E THE PROFILE NAME ON THE POOL STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid*} NOT FOUND.
- HCP1767E THERE IS MORE THAN ONE PROFILE NAME SPECIFIED ON THE PROFILE STATEMENT.
- HCP1768E AN MDISK OR MINIOPT STATEMENT WAS FOUND IN PROFILE *profname*.
- HCP1769E A[N] {INCLUDE|POOL} STATEMENT WAS FOUND IN PROFILE *profname*. {INCLUDE|POOL} STATEMENTS CANNOT BE IMBEDDED IN PROFILES.
- HCP1770E AN INCLUDE STATEMENT WAS NOT PRECEDED BY A USER OR IDENTITY STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}.
- HCP1771E THE PROFILE NAME IS MISSING FROM THE *type* STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid*}.
- HCP1772E MORE THAN ONE PROFILE NAME WAS SPECIFIED ON THE *type* STATEMENT FOLLOWING {USER *userid* | IDENTITY *userid*}.
- HCP1773E A PROFILE STATEMENT WAS FOUND PRECEDING THE DIRECTORY STATEMENT.
- HCP1774E THE RPWLST ENTRY - *rpw* - IS INVALID.
- HCP1775E PROFILE DEFINITION *profname* IS INVALID AND WILL NOT BE PROCESSED FOR ANY USER DEFINITIONS THAT INCLUDE IT.
- HCP1776E DEFAULT STORAGE SIZE EXCEEDS MAXIMUM STORAGE SIZE FOR {USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}
- HCP1779I THE DEVICE TYPE *type* ON THE PRECEDING CONSOLE STATEMENT IS INVALID AND SHOULD BE CHANGED TO THE APPROPRIATE VALUE. IT HAS BEEN DEFAULTED TO 3215 FOR THIS EXECUTION OF DIRECTXA.
- HCP1781E THE *option1* OPTION WAS SPECIFIED ON THE DIRECTXA COMMAND WITHOUT THE *option2* OPTION.
- HCP1782E THE DUPLICATE OPTION *option* WAS SPECIFIED ON THE DIRECTXA COMMAND.
- HCP1783E AN OPERAND OF THE SYSAFFIN STATEMENT IS MISSING.
- HCP1784E THE OPTION *option* SPECIFIED ON THE DIRECTXA COMMAND HAS AN INVALID *value* VALUE.
- HCP1785E THE {PROFILE *profname* | USER *userid* | POOL *poolid* | IDENTITY *userid* | SUBCONFIG *id*} FROM THE {PROFILE | USER | IDENTITY | SUBCONFIG} STATEMENT DOES NOT MATCH THE {PROFILE *profname* | USER *userid* | POOL *poolid* | IDENTITY *userid* | SUBCONFIG *id*} FROM THE INCLUDING LOAD STATEMENT.
- HCP1786E THE STATEMENT IS NOT VALID IN A CLUSTER DIRECTORY INDEX.
- HCP1787E THE DIRECTORY STATEMENT DOES NOT CONTAIN THE CPUID AND SYSAFNID FIELDS.
- HCP1788E THE CONFLICTING OPTION *option* WAS SPECIFIED ON THE DIRECTXA COMMAND.
- HCP1789E DUPLICATE SYSAFNID VALUE *value* WAS SPECIFIED ON THE SYSAFFIN STATEMENT.
- HCP1790E THE CPUID IS A DUPLICATE OF ONE FOUND ON A PREVIOUS DIRECTORY STATEMENT.
- HCP1791E THE LOAD STATEMENT WAS FOUND OUT OF ORDER.
- HCP1792E MORE THAN 56 SYSAFNIDS WERE SPECIFIED.

- HCP1793E AN OPERAND ON THE LOAD STATEMENT HAS AN INVALID VALUE - *value*.
- HCP1794E AN OPERAND ON THE SYSAFFIN STATEMENT HAS AN INVALID VALUE - *value*.
- HCP1795E POOL STATEMENT IS MISSING IN *fileid*.
- HCP1796E A POOL STATEMENT WAS FOUND FOLLOWING {USER *userid* | IDENTITY *userid*} IN *fileid*. THE CORRESPONDING LOAD STATEMENT DID NOT SPECIFY A POOL DEFINITION.
- HCP1797E CONFLICTING DEVICE INFORMATION HAS BEEN SPECIFIED ON MULTIPLE DIRECTORY CONTROL STATEMENTS CONTAINING THE SAME SYSTEM AFFINITY NAME - *sysafnid*.
- HCP1798E *operand* WAS SPECIFIED ON A(N) *stmt1* STATEMENT BUT THE *stmt2* STATEMENT WAS NOT SPECIFIED FOR {USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}.
- HCP1799E A(N) *stmt* STATEMENT WAS FOUND IN SUBCONFIG *id*. *stmt* STATEMENTS ARE NOT ALLOWED IN SUBCONFIG DEFINITIONS.
- HCP1799E A(N) *option* OPTION WAS FOUND IN SUBCONFIG *id*. *option* OPTIONS ARE NOT ALLOWED IN SUBCONFIG DEFINITIONS.
- HCP1879E *statement* STATEMENT WITH {WRKALLEG | NOWRKALLEG} OPTION FOLLOWING {USER *userid* | IDENTITY *userid* | SUBCONFIG *id*} DOES NOT APPLY TO VIRTUAL DEVICE *vdev* BECAUSE THE {PREVIOUS STATEMENT IS NOT AN MDISK | DEVICE IS DEFINED READ-ONLY}
- HCP1880E CLOSING QUOTATION MARK IS MISSING FROM QUOTED STRING FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}
- HCP1882E NICDEF REJECTED; {TYPE MUST BE SPECIFIED FIRST | DEVICE COUNT IS OUT OF RANGE n-nnnn | LAST DEVICE IS ABOVE FFFF} FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}.
- HCP2826E {SCPDATA | *operand*} CONTAINS INVALID UTF-8 DATA: *data* [FOLLOWING {PROFILE *profname* | USER *userid* | IDENTITY *userid* | SUBCONFIG *id*}]
- HCP2829E SCPDATA DOES NOT CONTAIN A VALID HEX STRING [FOLLOWING {IDENTITY *userid* | SUBCONFIG *id*}]

## Return codes

The DIRECTXA utility issues the following (RC) return codes:

### RC

#### Possible Causes

**0**

DIRECTXA processed successfully. The real user directory has been updated (unless the EDIT option was specified).

**1**

A directory source file was not found on an accessed disk.

**2**

An error was encountered while processing a directory source file.

**3**

A invalid option was specified on the DIRECTXA command line.

**4**

No errors were encountered while processing a directory source file, but one or more of the following conditions exists:

- You do not have the proper privilege class to update the real user directory. To determine if your virtual machine has the proper privilege class to update the real user directory, issue the following CP command and look for the string "DIAG3C" in the response:

```
cp query commands
```

- You have attempted to update the real user directory with an SSI-enabled source directory, which is not valid in the current system configuration.



- You have attempted to update the real user directory with a directory that is not SSI-enabled, which is not valid in the current system configuration.

**5**

Condition code 1 was received from DIAGNOSE X'3C', which means a class A, B, or C user updated a virtual directory.

**6**

Condition code 2 was received from DIAGNOSE X'3C', which means a invalid directory pointer was found in the volume label.

**7**

Condition code 3 was received from DIAGNOSE X'3C', which means a fatal I/O error occurred.

**9**

The directory has been rewritten, but warning messages have been issued.

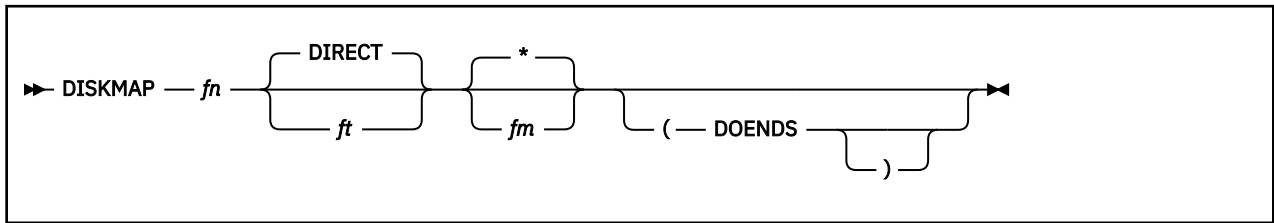
**> 100**

Return codes greater than 100 may be returned accompanied by message 764—except for code 333, which will not be accompanied by message 764. See the explanation for message 764 for details on these return codes.

**333**

DIRECTXA was run in EDIT mode, and at least one password was changed to NOLOG.

## DISKMAP



### Purpose

Use the DISKMAP utility to summarize the MDISK statements in the user directory. The output produced by DISKMAP shows gaps and overlaps between minidisk assignments.

DISKMAP does NOT replace the EDIT function of the DIRECTXA command. You should use both to check your directory after changes.

Also see the [DIRMAP utility](#) in *z/VM: CMS Commands and Utilities Reference*.

### Operands

#### *fn*

is the file name of the directory to be mapped.

#### *ft*

is the file type of the directory to be mapped. The default is DIRECT.

#### *fm*

is the file mode of the directory to be mapped. The default is an asterisk (\*).

#### DOENDS

causes MDISK statements that had "END" specified for the ending cylinder/block in the directory being mapped to be included in the DISKMAP output file.

### Responses

The output from DISKMAP is a file sent to your A-disk. The file name of the output is the same as that of the target directory. The file type of the output is DISKMAP. The file contains information on MDISK statements found in the directory. The files are in order by volume in the output file. Gaps between minidisks and overlapping minidisks are flagged.

### Usage Notes

1. By default, the DISKMAP utility is located on the cross release utilities disk (PMAINT 551). This disk must be accessed to run the DISKMAP utility.
2. Because some DASD types come in several sizes, DISKMAP does not list gaps found after all minidisks. You need to know the maximum cylinder/block value for your DASD type.
3. DISKMAP creates the map on your A-disk. If your directory is very large and your A-disk is almost filled, you may need to find some extra disk space in order to run DISKMAP.
4. You may choose to include some overlaps in the directory. DISKMAP flags *all* overlaps; you must understand your layout to determine if a particular overlap is expected or in error.
5. If the directory contains MDISK statements with "END" specified as the ending cylinder/block, DISKMAP ignores these MDISK statements and issues a warning message. In order to have these MDISK statements appear in the DISKMAP output file, you need to invoke DISKMAP with the "DOENDS" option.
6. DISKMAP does not allow an asterisk (\*) as an input parameter for a file type.

7. If DEVNO is specified as a starting cylinder for a minidisk statement, the statement will be ignored and a single message for all statements will be issued explaining this. If T-DISK or V-DISK is specified as a starting cylinder, those statements will not be included and no message will be issued.
8. From an authorization perspective, the USERID value specifies the user ID to which the MDISK belongs. For multiconfiguration virtual machine instances, the MDISKS can be defined within either the identity or subconfiguration entries. In both cases the USERID value specifies the name from the IDENTITY statement. If the SUBCONFIG statement is not referenced by any BUILD statement, the USERID value contains a value of \*NOUSER!.
9. The SUBCONFIG value in the output is blank unless the MDISK was defined within a subconfiguration entry.
10. The MEMBER value in the output is \* unless the MDISK was defined within a subconfiguration entry, in which case the value displayed is the system identifier from the BUILD statement. If there is no corresponding BUILD statement, this value is blank.
11. DISKMAP does not perform stringent directory verification. It is assumed that all input directories are verified using DIRECTXA to be valid directories.

## Examples

If the directory contains the following statements:

```

DIRECTORY SSI 123 3390 M01RES M02RES M03RES
*
*****
*
USER $DIRECT$ NOLOG
  MDISK A01 3390 001 020 M01RES R
  MDISK A02 3390 001 020 M02RES R
*
*****
*
*
IDENTITY MAINT      MAINT      128M 1000M ABCDEFG
  BUILD ON TAPE1 USING SUBCONFIG MAINT-1
  BUILD ON TAPE2 USING SUBCONFIG MAINT-2

SUBCONFIG MAINT-1
  MDISK CF1 3390 039 120 M01RES RR READ      WRITE      MULTIPLE
  MDISK CFD 3390 159 001 M01RES RR READ      WRITE      MULTIPLE
* END OF MAINT-1

SUBCONFIG MAINT-2
  MDISK CF1 3390 039 120 M02RES RR READ      WRITE      MULTIPLE
  MDISK CFD 3390 159 001 M02RES RR READ      WRITE      MULTIPLE
* END OF MAINT-2

```

DISKMAP will provide the following minidisk summary in the output file:

```

-----
VOLUME  USERID      CUU  DEVTYPE  START      END      SIZE      GAP  SUBCONFIG  MEMBER
M01RES  $DIRECT$      A01   3390    00001      00020    00020    GAP
                MAINT         CF1   3390    00039      00158    00120    GAP  MAINT-1    TAPE1
                MAINT         CFD   3390    00159      00159    00001    MAINT-1    TAPE1
-----

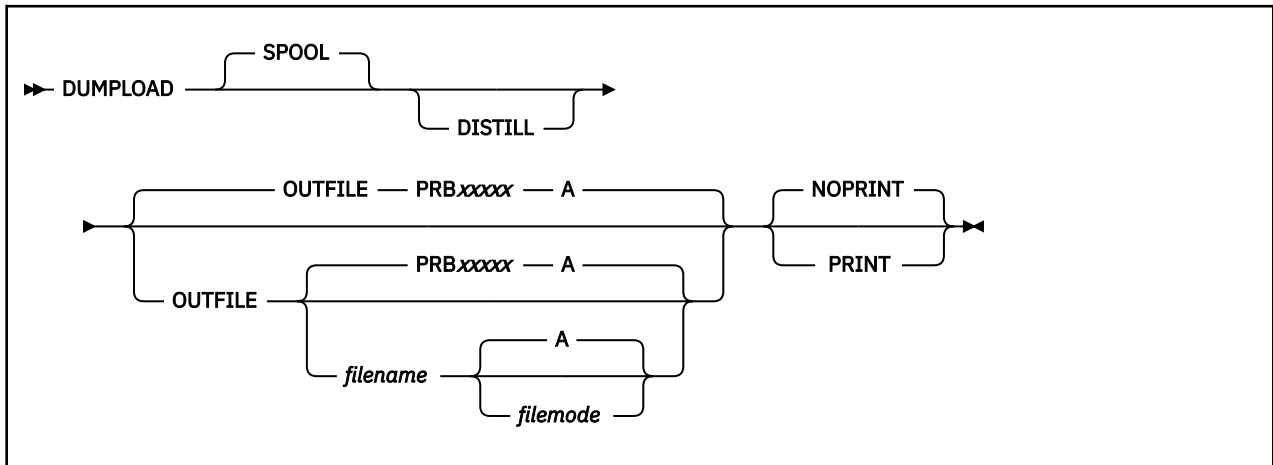
```

```

-----
VOLUME  USERID      CUU  DEVTYPE  START      END      SIZE      GAP  SUBCONFIG  MEMBER
M02RES  $DIRECT$      A02   3390    00001      00020    00020    GAP
                MAINT         CF1   3390    00039      00158    00120    GAP  MAINT-2    TAPE2
                MAINT         CFD   3390    00159      00159    00001    MAINT-2    TAPE2
-----

```

## DUMpload



### Purpose

Use the DUMpload utility to process system abend dumps, stand-alone dumps, and virtual machine dumps. Issue DUMpload to:

- Load a system abend dump from a reader spool file or a reader system data file into a CMS dump file.
- Load a virtual machine dump from a reader system data file into a CMS dump file.

### Operands

#### SPool

indicates the dump to be processed is in a reader spool file or reader system data file. SPOOL is the default.

Before issuing DUMpload, be sure that your reader is at address X'00C', and it is spooled for the correct class for the type of dump you want to process. System abend dumps are class D and virtual machine dumps are class V. Also, be sure that the dump you want to process is in HOLD NONE status. Use the ALL option of the QUERY READER command to display the HOLD status of your reader files as well as other information about dump files in your reader. Also, the ORDER READER command can be used to place the dump you want processed as the first file in the reader queue.

Also make sure that your user ID has the correct privilege class to load a CP dump. To read a CP system abend dump, you need class C or class E authorization. For a virtual machine dump, you need class G authorization. If a dump file exists under a different user ID, ask your operator to transfer that file to your user ID so that your virtual reader can access it.

Each invocation of DUMpload will process only one dump. DUMpload will read the first available reader file that is a dump, matches the specified spool class, and is not in HOLD status. If that dump is of the same release as DUMpload, DUMpload asks you if you want DUMpload to process it. If you answer "NO", DUMpload will terminate without processing the file. If you answer "YES", DUMpload will process the file and then terminate. In either case, the reader file that DUMpload read will now be in USER HOLD status and must be changed to HOLD NONE if you wish DUMpload to process it again.

If the reader is spooled class \*, DUMpload will attempt to find the first available class D system abend dump. If no available class D file is found, DUMpload will attempt to find the first available class V virtual machine dump.

#### DISTILL

extracts an abridged dump from a hard abend dump or a snap dump on a spool file. This option is ignored if the input dump is not a hard abend or a snap dump. An abridged dump is typically much smaller and easier to transmit to IBM support personnel for diagnosis. If the input dump contains a

frame table, this operand causes a correlation table to be created from the frame table and stored in the loaded dump. For both frame table dumps and correlation table dumps, pageable PGMBK pages are omitted if the original dump was taken by using the PGMBKS ALL option. If the original dump was taken by using the PGMBKS FAILUSER option, those PGMBK pages are retained.

The original dump remains on the spool file, and if necessary, the DUMPLD2 utility, or the DUMpload utility without the DISTILL option, can be used later to load the complete dump for transmission to IBM.

### OUTFILE

is a keyword for specifying the file name and file mode of the CMS dump file to be created. You can specify only one CMS dump file on the command line.

### PRBxxxxx

is the default file name of the CMS dump file, where xxxxx is a five-character numeric from 00000 to 99999. The dump is given a file type of DUMP nnnn, where nnnn is 0001 through 9999.

### filename

is the file name of the CMS dump file to be created. It can be any valid CMS file name, but cannot be the same as a DUMpload operand.

### filemode

is any file mode to which you have write access. The default is file mode A.

### PRINT

### NOPRINT

are allowed for compatibility. The PRINT function formerly used to print VMDUMP command dumps of ESA/390 guests is now in the PRTDUMP command of the Dump Viewing Facility. For any other dumps, use the VM Dump Tool to create a DUMPLOG file that can then be printed. See Usage Note [“5” on page 2181](#).

## Usage Notes

1. You can use the CMS dump files you create with the DUMpload command as input for the Dump Viewing Facility or the VM Dump Tool.
2. If you use DUMpload to read a system abend dump or virtual machine dump from a reader spool file or system data file, the system saves the file upon completion of the command processing. In addition the reader is also closed at the beginning as well as at the end of the command processing.
3. All the CMS dump files created from a single virtual machine dump have the same file name. The file type of the first is *DUMP0001*, the second *DUMP0002*, and so on. When several CMS files are created for a virtual machine dump, the file names of related files are put into the CMS dump files. Therefore, you should not rename those files except as a group. Instead, use the OUTFILE option on the DUMpload command to give the set of files the file name you prefer.
4. If DUMpload runs out of space on your disk or file space during the processing of a virtual machine dump for several address spaces, DUMpload stops but does not erase the dump files it has created, including an incomplete one. You can use the Dump Viewing Facility to view the files you do have, or obtain more space and reissue the DUMpload command. It will start from the beginning and recreate all files as if it had never been run. If you do not erase the files you do have, you must specify another file name on the DUMpload command.
5. Even though the printing function formerly in the DUMpload command is now in the PRTDUMP command, the PRINT and NOPRINT options are still allowed for compatibility. If PRINT is specified, DUMpload issues a message informing you that the dump should be printed using the PRTDUMP command. DUMpload ignores NOPRINT.
6. Before you enter the DUMpload command, be sure that you have sufficient space at a read/write file mode or enough SFS file space to hold the CMS dump. Forty-two hundred 4K blocks of SFS file space are needed for each 16 MB of dumped storage. The following chart indicates how many cylinders for CKD/ECKD or blocks for FBA devices of free DASD space you will need for each 16 MB of dumped storage, based upon the device type and the block size of the formatted minidisk. For dumps of less than 16 MB, the number of cylinders or blocks is proportional.

## DUMpload

BLKSIZE	3380	3390	FB-512
512	34	36	52,700
1024	36	34	48,960
2048	31	27	38,880
4096	28	23	33,120

The DASD space requirement shown assumes the dump was produced using a single invocation of VMDUMP. The CMS blocks listed is also the number of free CMS blocks of each size (512, 1024, etc.). When VMDUMP operates in continuous output mode, and page ranges are not dumped in strict ascending sequence, DUMpload must allocate additional DASD space for one or more work files, named xxxxxxxx \$DUMPLD\$, in order to create a merged output file.

Be sure the space allocated for the dump file allows for discontinuous saved segments, if they were requested with the dump. If the dumped address ranges include unreferenced (never used) storage, those pages are excluded, decreasing the storage requirement. Use the number of records in the VMDUMP spool file, which you can obtain using the QUERY RDR command, to estimate the amount of storage actually dumped. The size of VMDUMP spool file records is 4 KB.

7. Use the DUMPLD2 utility as an alternative to DUMpload, to segment a dump into multiple files in order to load or transmit it, especially for larger dumps. The utility will handle all dump types currently supported by DUMpload, except for multiple address space dumps.

### Messages

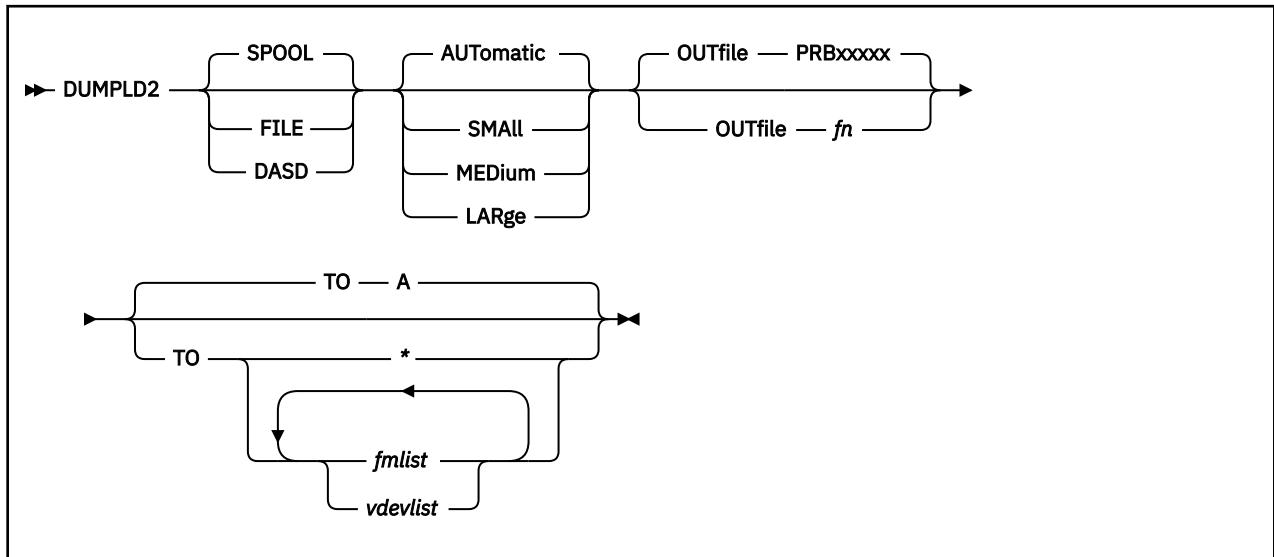
- HCP002E Invalid operand - *operand*
- HCP850I UNABLE TO READ DUMP
- HCP852E FATAL I/O ERROR WRITING DUMP, FSWRITE|FSERASE|FSTATE| CMSCALL RC = *rc*
- HCP853E NO DUMP FILES EXIST
- HCP8150I PROCESSING THE DUMP *fileid*
- HCP8156A DO YOU WANT TO PROCESS THIS DUMP? (YES/NO)
- HCP8160I DUMP FROM *code (dumptype) nodeid product release*
- HCP8161E CONFLICT BETWEEN OPERANDS *operand* AND *operand*
- HCP8162 FATAL I/O ERROR READING DUMP, FSREAD RC = *rc*
- HCP8164E DUMP FILE NAME *filename* IS NOT A VALID CMS FILE NAME
- HCP8167E *operand* OPERAND IS NOT VALID FOR A VIRTUAL MACHINE DUMP
- HCP8167I VIRTUAL MACHINE DUMP FROM *product release*
- HCP8168I VIRTUAL MACHINE DUMP, FORMAT = *vmtype*, DUMPID = *dumpid*
- HCP8169I DUMP IS NOT COMPLETE
- HCP8169I DUMP IS NOT COMPLETE; IT CONTAINS NO CP STORAGE
- HCP8175E REQUIRED SYSTEM STORAGE NOT AVAILABLE
- HCP8177E DUMP FILE *filename* ALREADY EXISTS ON *filemode* DISK
- HCP8178E DUMP FILE MODE *filemode* INVALID
- HCP8179I THE OPTION PRINT IS NOT SUPPORTED BY THE DUMpload COMMAND
- HCP8180E DUMpload FAILED BECAUSE NO R/W *filemode* DISK IS AVAILABLE
- HCP8181I PLEASE REFER TO THE DUMP VIEWING FACILITY PRDUMP COMMAND FOR EQUIVALENT FUNCTION
- HCP8182E UNKNOWN DFIR FORMAT
- HCP8183I DUMpload *product release*

- HCP8184E RELEASE OF DUMPLOAD AND DUMP ARE NOT COMPATIBLE
- HCP8185I THE NUMBER OF ALTERNATE CPUS FOUND EXCEEDS THE MAXIMUM ALLOWED
- HCP8186I THE OPTION DISTILL IS IGNORED; DUMP IS NOT A HARD ABEND OR SNAP DUMP
- HCP8187I DUMP TAKEN WITH FRMTBL {YES | NO} PGMBKS {ALL | NONE | FAILUSER}
- HCP8188I DUMP WILL BE DISTILLED TO FRMTBL NO PGMBKS {NONE | FAILUSER}

You can receive CP or CMS error messages other than those indicated here. Any CMS errors that occur during DUMPLOAD processing are reported by CMS through CMS messages or CMS macro return codes. All prompts and CP error indications are reported through CP messages.

You can find a description of the message you receive in [\*z/VM: CP Messages and Codes\*](#).

## DUMPLD2



### Purpose

Use the DUMPLD2 utility as an alternative to DUMpload, to segment a dump into multiple files in order to load or transmit it. The utility will handle all dump types currently supported by DUMpload, except for multiple address space dumps. Hard Abend, Soft Abend, SNAPDUMP, and VMDUMP will be handled. Stand-alone dumps will also be handled.

### Operands

#### SP00L

indicates dump to be processed is in spool. This is the default.

#### FILE

indicates the dump to be processed is a single CMS file on disk. If you specify FILE, you must enter the file name of the dump and file mode where it can be found. The file type is assumed to be 'DUMP0001'.

#### DASD

indicates the dump to be processed is on DASD and was written by stand-alone dump. You will be prompted to enter the virtual device number of the first device containing the dump. DUMPLD2 will read data from that device and display information about which devices contain the dump. All of those devices need to be defined, attached, or linked, as appropriate, before DUMPLD2 will process the dump.

The DUMPLD2 utility requires all of the real devices that are part of the stand-alone dump configuration to be attached at the same virtual device numbers that were used when SDINST formatted them. If this is not the case, error messages are displayed and the dump is not loaded. To help you set up the devices correctly, the DUMPLD2 utility displays a device table which has a DUMPLD2 VDEV column that shows the virtual device at which the real device must be attached. You can use the CP DEFINE command to change the virtual device number if the device is attached at the wrong number.

If the dump was written to SCSI DASD, you must respond with the virtual device number of the first emulated device (EDEV). Additional dump DASD must be defined at sequential virtual device numbers starting after the first number specified by the user.

#### AUTomatic

indicates that the record count of each output file will be determined by the sizes of the disks specified by the user. The record count will be calculated based on the disk space available on the



specified file modes and virtual device addresses. File modes and virtual device addresses that have less than 65,536 (the 'small' record count) available will be ignored. The smallest available space on any of the remaining file modes or virtual device addresses will be compared to the "small" "medium" or "large" values. The largest of these 3 sizes to fit in that space will be chosen as the record count for each of the CMS files created. This is the default if the record count parameter is omitted.

**SMALL**

indicates the record count of each file will be:

65,536 records, which is approximately .25 G, or 365 cylinders on a 3390 disk.

**MEDium**

indicates the record count of each file will be:

262,144 records, which is approximately 1 G, or 1460 cylinders on a 3390 disk.

**LARge**

indicates the record count of each file will be:

1,048,576 records, which is approximately 4 G, or 5825 cylinders on a 3390 disk.

**OUTfile *fn***

is a keyword for specifying the file name for the CMS files (of file type MDMPxxxx) to be created. If *fn* is omitted, the utility will produce an error message and exit.

**Note:** If 'OUTFILE' is omitted, the multiple dump files will have a file name of format "PRBxxxxx", where "xxxxx" is an unused number from 00000 to 99999, chosen by the utility.

**TO**

is a keyword indicating the disks on which the CMS files should be created (of file type MDMPxxxx). The user may specify file modes or virtual device addresses, or a combination in any order. If the user does not specify any file modes or virtual device addresses after 'TO', the utility will display an error message and exit.

**Note:** If 'TO' is omitted, the MDMPxxxx files will be written to the A disk.

***fmlist***

indicates the set of file modes on which to create the CMS files (MDMPxxxx). Accepted inputs include:

- a blank delimited list: "A D E F"
- a range: "A-F", indicating all R/W file modes in that range are to be used.

If a specified file mode is not accessed or is not R/W, it will be ignored and not used.

***vdevlist***

indicates the set of virtual devices on which to create the CMS files. Accepted inputs will include:

- a blank delimited list of 3 or 4-digit hex values: "0122 125 0121"
- a range: "0121-012E" (3 or 4 digit hex values are accepted)

If a specified virtual device is not attached as a CMS DASD, or is not R/W, it is ignored and not used.

**Note:** The user must have at least one free file mode available so that a virtual device may be accessed if it is not currently accessed.

**\***

indicates that the utility will search for available R/W file modes and use them in order of search sequence.

**Usage Notes**

1. You can use the CMS files you create with DUMPLD2 as input for the VM Dump Tool.
2. If you use DUMPLD2 to read a dump from a reader spool file, the system changes the file to User 'HOLD' status and does not purge it upon completion of the command processing.

3. If DUMPLD2 runs out of file modes or virtual devices during the processing of a dump, DUMPLD2 stops but does not erase the CMS files it has created, including a possibly incomplete one. You can use the VM Dump Tool to view the files you do have, or you can obtain more space and reissue the DUMPLD2 command. It will start from the beginning and recreate all files as if it had never been run – the existing files will be overwritten. If you wish to keep the old files, specify another file name when re-invoking DUMPLD2.
4. After the 'TO' operand, a combination of *fmlist*, *vdevlist* and '\*' may be intermixed in any order.  
For example, "D 0120-012F A-C \*" is valid.
5. You can specify *fmlist* or *vdevlist* for as many file modes or VDEVs as you have – duplicate values will be ignored.
6. Every CMS file that DUMPLD2 produces will have the file type of "MDMPxxxx".
7. If you are processing a Hard Abend dump, the frame table must not span over the first two different unaccessed virtual device addresses.
  - If the first disk specified is an unaccessed VDEV, the frame table must be contained within the multiple files on that VDEV.
  - If the first few disks specified are file modes, the frame table must be contained within the multiple files on these file modes and must not span to an unaccessed VDEV.
8. If one of the file modes specified is an SFS directory, multiple dump files are loaded up to the space limits for that directory. The utility will not calculate in advance how many files will be written or if there will be sufficient space to load the entire dump.
9. 'VMLINK' is used to temporarily access virtual device addresses. The file modes used to temporarily access virtual device addresses are chosen according to the default settings in 'VMLINK'.
10. The order in which file modes, virtual device addresses, or SFS directories are specified after the 'TO' operand is the order to which they are written.

## Messages

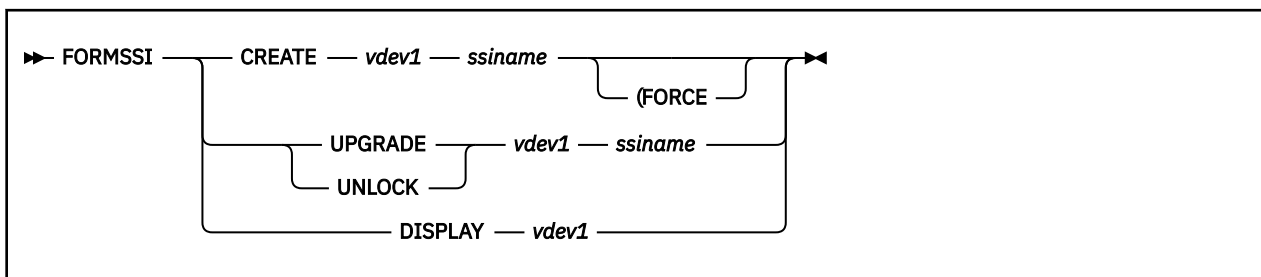
- HCP002E Invalid operand *operand*
- HCP0003E Invalid option - command contains extra option(s) starting with *operand*
- HCP0026E Operand missing or invalid
- HCP022E A virtual device number was not supplied or it was invalid.
- HCP1008E Device *userid vdev* is not a DASD
- HCP040E Device *vdev* does not exist
- HCP8164E Dump file name *fn* is not a valid CMS file name
- HCP8170E There are no default file names available
- HCP8178E Dump file mode *filemode* invalid
- HCP8210I *numfiles fn* MDMPxxxx file(s) will be created on disk *filemodenumfiles fn* MDMPxxxx file(s) will be created on VDEV *vdev*
- HCP8211E Please reset SPOOL READER class to V, D or \*
- HCP8212E No unheld class V or class D files exist in SPOOL
- HCP8213I Created *fileid*
- HCP8214I Duplicate values will be ignored *duplicates*
- HCP8215I DUMPLD2 cannot determine how many files will be written to the SFS directory specified
- HCP8216I *n* files have been written to the SFS directory at file mode – *mode*
- HCP8217E Error in erasing incomplete file – ERASE rc = *rc*Error in copying incomplete file to next disk – ERASE rc = *rc*
- HCP8218E Incorrect mode *mode* in range - *range*
- HCP8219E VDEV must be a hex value - *vdev*

- HCP8220E VDEV must be 3 or 4 digits - *vdev*
- HCP8221E Incorrect range specified - *range*
- HCP8223E File mode *mode* is not accessed
- HCP8224E File mode *mode* is not R/W
- HCP8225E Device *vdev* does not exist
- HCP8226E Device *vdev* is not formatted as CMS DASD
- HCP8227E Device *vdev* is not R/W
- HCP8228E Insufficient space for any files – *device*
- HCP8229E Error in writing to file *fileid* – EXECIO rc = *rc* Error in closing file *fileid* – FINIS rc = *rc* Error in opening file *fileid* – EXECIO rc = *rc*
- HCP8230E Insufficient disk space to continue.. Exiting
- HCP8231E Error in accessing disk – VMLINK rc = *rc* Error in releasing disk – VMLINK rc = *rc*
- HCP8232E File name must be specified with the OUTFILE operand
- HCP8234E *Fn DUMP0001 mode* not found
- HCP8235E No usable devices/disks specified
- HCP8236E Must specify file modes or VDEVs after the TO operand
- HCP8237I TO operand not specified. MDMPxxxx files will be created on A disk
- HCP8238I System dump not found. Searching for VMDUMP
- HCP8239A Dump File Name?
- HCP8240A Dump File Mode?
- HCP8241I Automatic file size chosen – *filesize* records
- HCP8242I Insufficient disk space to create the remaining *n* files
- HCP8243I Dump is not complete: *records* records written of *totalrecords* expected
- HCP8243I Dump not complete when it was created; it contains no CP storage. Only *number* records were copied from the SPOOL file and written to the CMS file.
- HCP8243I Dump not complete when it was created; it contains some CP storage. *RecsWritten* of expected *number* records were dumped.
- HCP8243I Dump not complete when it was created; it contains some CP storage. *RecsWritten* records written.
- HCP8244E DUMPLD2 failed because no R/W File Modes are available
- HCP8245E File mode incorrect – *mode*
- HCP8246E DUMPLD2 cannot create more than 9999 files from one dump
- HCP8247E HCPxxx MODULE was not found so it could not be used to determine whether the dump is complete.
- HCP8248E QUERY READER spoolid and DIAGNOSE X'14' spoolid mismatch. Make sure the dump spool file is ORDERed to top of READER queue.
- HCP8249I The dump is larger than what QUERY READER command shows. It really has *nnnnnnnnnn* records.
- HCP8277E Pipe error *rc* processing the dump.
- HCP8278E Enter virtual device number of first dump device.
- HCP8281E Device *vdev* has unexpected format: *format*
- HCP8282E Device *vdev* appears to be formatted as an additional stand-alone dump DASD and not as the IPL DASD
- HCP8283E Error *rc* during function
- HCP8284E Format of *item* on device *vdev* was not as expected.

## DUMPLD2

- HCP8285E Error validating dump data - *data description* Reason code *rc*
- HCP8286E Device *vdev* does not contain a dump
- HCP8287E Generated VDEV past maximum allowed VDEV
- HCP8288E Pipe error, *rc=rc*, while putting records into the pipe
- HCP8289E No output stream for pipe
- HCP8290E Unexpected error. Location *number*

# FORMSSI



## Purpose

Use the FORMSSI utility to create, update, unlock, or display a persistent data record (PDR) on cylinder zero of a disk.

## Operands

### CREATE

specifies that a PDR is to be created. Any existing PDR that does not appear to be in use is overwritten.

### *vdev1*

specifies the virtual device number representing the dedicated volume, full-pack minidisk, or regular minidisk on which the PDR is to be created or modified. The disk must be a 3390 or on a 3390 DASD.

### *ssiname*

specifies the name of the single system image (SSI) cluster. The name can be up to eight characters long and can contain the following characters:

- A-Z
- 0-9

### UPGRADE

specifies that an existing PDR is to be upgraded (for example, when a new release of z/VM is to be installed).

### UNLOCK

specifies that an existing PDR that was left locked by a previous unsuccessful invocation of the FORMSSI utility is to be unlocked.

### DISPLAY

specifies that an existing PDR SSI cluster name and state are to be displayed.

## Options

### FORCE

specifies that a PDR is to be created even if the target device contains a PDR that might be in use.

## Usage Notes

1. By default, the FORMSSI utility is located on the cross release utilities disk (PMAINT 551). This disk must be accessed to run the FORMSSI utility.
2. No special privileges are required to run the FORMSSI utility. However, the utility issuer must have read/write access to the volume on which the PDR resides when using any operand other than DISPLAY. The issuer must have read access to the volume on which the PDR resides when using DISPLAY.
3. The PDR structure allows complete forward and backward compatibility among different releases of z/VM. This compatibility allows the sizes of record structures to increase over time using the

UPGRADE function. However, the UPGRADE function does not accommodate decreasing these structure sizes, and produces an error message if such a change is attempted. z/VM will upgrade the PDR automatically during IPL if the system being IPLed determines that the record structure supported by the system is larger than the record structure in the PDR on disk.

4. The PDR resides on cylinder zero and its location is compatible with the CP allocation record and other data that might also reside on cylinder zero.
5. The PDR must reside on a volume that has been allocated for CP use (through CPFMTXA) and cylinder zero must be allocated as PERM.
6. During system installation, a full-pack minidisk is defined for the DASD volume where the PDR is created. That minidisk address is 141 and is owned by user PMAINT.
7. The "last change:" field in the FORMSSI DISPLAY output will equal a constant of \$LCLSYS\$ when a member is attempting to join the SSI cluster. When set to \$LCLSYS\$, the last change to the current slot was made by the member in that slot.

## Examples

Example 1:

Querying an active PDR on virtual device EFE0 displays the following information:

```
formssi display efe0
HCPPDF6618I Persistent Data Record on device EFE0 (label VMCOM1) is for CLUSTERA
HCPPDF6619I PDR state: Unlocked
HCPPDF6619I time stamp: 04/11/15 21:22:03
HCPPDF6619I cross-system timeouts: Enabled
HCPPDF6619I replacement: None
HCPPDF6619I PDR slot 1 system: VMSYS01
HCPPDF6619I state: Joined
HCPPDF6619I time stamp: 04/11/15 21:22:00
HCPPDF6619I last change: VMSYS01
HCPPDF6619I PDR slot 2 system: VMSYS02
HCPPDF6619I state: Joined
HCPPDF6619I time stamp: 04/11/15 21:21:40
HCPPDF6619I last change: VMSYS02
HCPPDF6619I PDR slot 3 system: VMSYS03
HCPPDF6619I state: Joining
HCPPDF6619I time stamp: 04/11/15 21:21:57
HCPPDF6619I last change: VMSYS03
HCPPDF6619I PDR slot 4 system: VMSYS04
HCPPDF6619I state: Down
HCPPDF6619I time stamp: 04/02/15 17:02:25
HCPPDF6619I last change: VMSYS02
```

Example 2:

Querying an obsolete PDR on virtual device EFEF displays the following information:

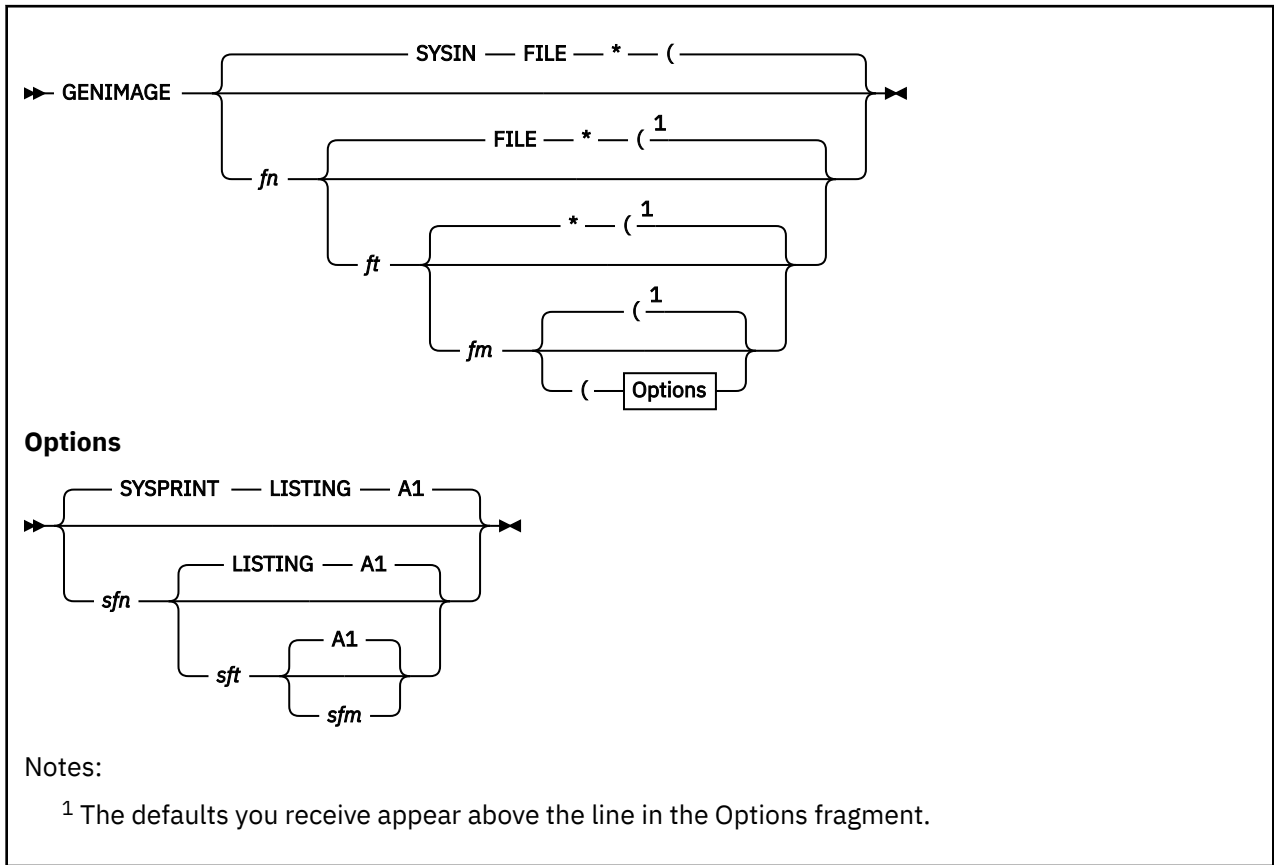
```
formssi display efef
HCPPDF6618I Persistent Data Record on device EFEF (label VMCOM1) is for CLUSTERA
HCPPDF6619I PDR state: Obsolete
HCPPDF6619I time stamp: 04/11/15 21:22:03
HCPPDF6619I replacement: EFE0
```

## Messages

- HCP1635I Persistent data record (PDR) on volume *label* for SSI *ssiname* automatically upgraded
- HCP6601E No function specified
- HCP6601E No Persistent Data Record device number specified
- HCP6601E No SSI name specified
- HCP6602E Function *name* not recognized
- HCP6603E "*name*" is not a supported SSI name
- HCP6603E "*vdev1*" is not a supported Persistent Data Record device
- HCP6604E Parameter "*name*" not expected

- HCP6605E Error *number* reading Persistent Data Record device label
- HCP6605E Error *number* reading/writing Persistent Data Record
- HCP6606E Device *vdev1* is not CP-formatted or cylinder zero is not allocated PERM
- HCP6607E Device *vdev1* does not contain a Persistent Data Record
- HCP6608E Persistent Data Record on device *vdev1* is locked, possibly by *system*
- HCP6609E Existing Persistent Data Record cannot be upgraded because a newer PDR version was found on device *vdev1*
- HCP6610E Device *vdev1* contains Persistent Data Record for *ssiname*
- HCP6611E Persistent Data Record on device *vdev1* not locked
- HCP6612E Persistent Data Record on device *vdev1* for *ssiname* may be in use
- HCP6613R Device *vdev1* label is *label* - continue (Yes/No)?
- HCP6614I Persistent Data Record created on device *vdev1*
- HCP6615I Persistent Data Record upgraded on device *vdev1*
- HCP6616I Persistent Data Record unlocked on device *vdev1*
- HCP6617I Persistent Data Record on device *vdev1* not changed
- HCP6618I Persistent Data Record on device *vdev1* (label *label*) is for *ssiname*
- - HCP6619I PDR                   state: *state1*
  - HCP6619I                   time stamp: *timestamp1*
  - HCP6619I cross-system timeouts: {Enabled|Disabled}
  - HCP6619I                   replacement: {None|*valid*}
  - HCP6619I PDR slot *slot*   system: *system1*
  - HCP6619I                   state: *state2*
  - HCP6619I                   time stamp: *timestamp2*
  - HCP6619I                   last change: *system2*
- HCP6620E Persistent Data Record on device *vdev1* is obsolete
- HCP9052W Error upgrading PDR on volume *label* for SSI *ssi\_name*

# GENIMAGE



## Purpose

Use the GENIMAGE utility to create text decks containing the following:

- Forms control buffer (FCB) images
- Copy modifications
- Character arrangement tables (CATs)
- Graphic character modifications
- Library character sets.

## Operands

**fn**

identifies the file name of the CMS input control file. The name may be up to eight alphanumeric characters long. If a name is not specified, the default file name is SYSIN.

**ft**

identifies the file type of the CMS input control file. The file type may be up to eight alphanumeric characters long. If a file type is not specified, the default file type is FILE.

**fm**

**\***

identifies the file mode of the CMS input control file. The file mode may be up to two alphanumeric characters long. If a file mode is not specified, the default file mode is \*.



**sfn**

identifies the file name of the file into which GENIMAGE places the message listing. If a name is not specified, the default file name is SYSPRINT.

**sft**

identifies the file type of the file into which GENIMAGE places the message listing. If a type is not specified, the default file type is LISTING.

**sfm**

identifies the file mode of the file into which GENIMAGE places the message listing. If a file mode is not specified, the default file mode is A1.

Below are listed the “starter set” of character sets supplied by IBM for both the 3800 Model 1 and Model 3 printers. They are supplied with VM, and they all have names of the form XTB1xxxx TEXT. They are built during the system generation process.

AN	FM12	GU10	ODA	RN
AOA	FM15	GU12	ONA	SN
AOD	GF10	GU15	ONB	TN
AON	GF12	G11	PCAN	TU10
A11	GF15	HN	PCHN	T11
BOA	GN	H11	PN	XN
BON	GS10	KN1	P11	YN
DUMP	GS12	OAA	QN	2773
FM10	GS15	OAB	QNC	2774

Below are listed the starter set of character sets supplied by IBM for the 3800 Model 3 printer only. They are supplied as both text and source files on the S-disk, and they all have names of the form XTB1xxxx TEST S. They are built during the system generation process.

AE10	EITR	GP12	LR12	SB12
BITR	ESTR	GR10	OB10	SI10
BRTR	GB10	GSC	OR10	SI12
CE10	GB12	GT10	PB12	SO12
CE12	GF10	GT12	PI12	SR12
CO10	GF12	GT15	PR10	ST10
CR10	GF15	GUC	PR12	ST12
DOTR	GFC	LB12	RT10	ST15
EBTE	GI12			

These fonts are described in the *IBM 3800 Printing Subsystem Programmer's Guide*.

The GENIMAGE utility produces two groups of output files. The first is the listing file (*fileid sfn sft sfm* as above) that contains the messages from the GENIMAGE utility. The second is a TEXT deck for each NAME statement in the input file. Below are listed the meanings of the file names of these TEXT decks.

**XTB1xxxx**

For a character arrangement table

**FCB3xxxx**

For a forms control buffer

**MOD1xxxx**

For a copy modification module

## GENIMAGE

### **GRAFxxxx**

For a graphic character modification module

### **LCS1xxxx**

For an image library character set module

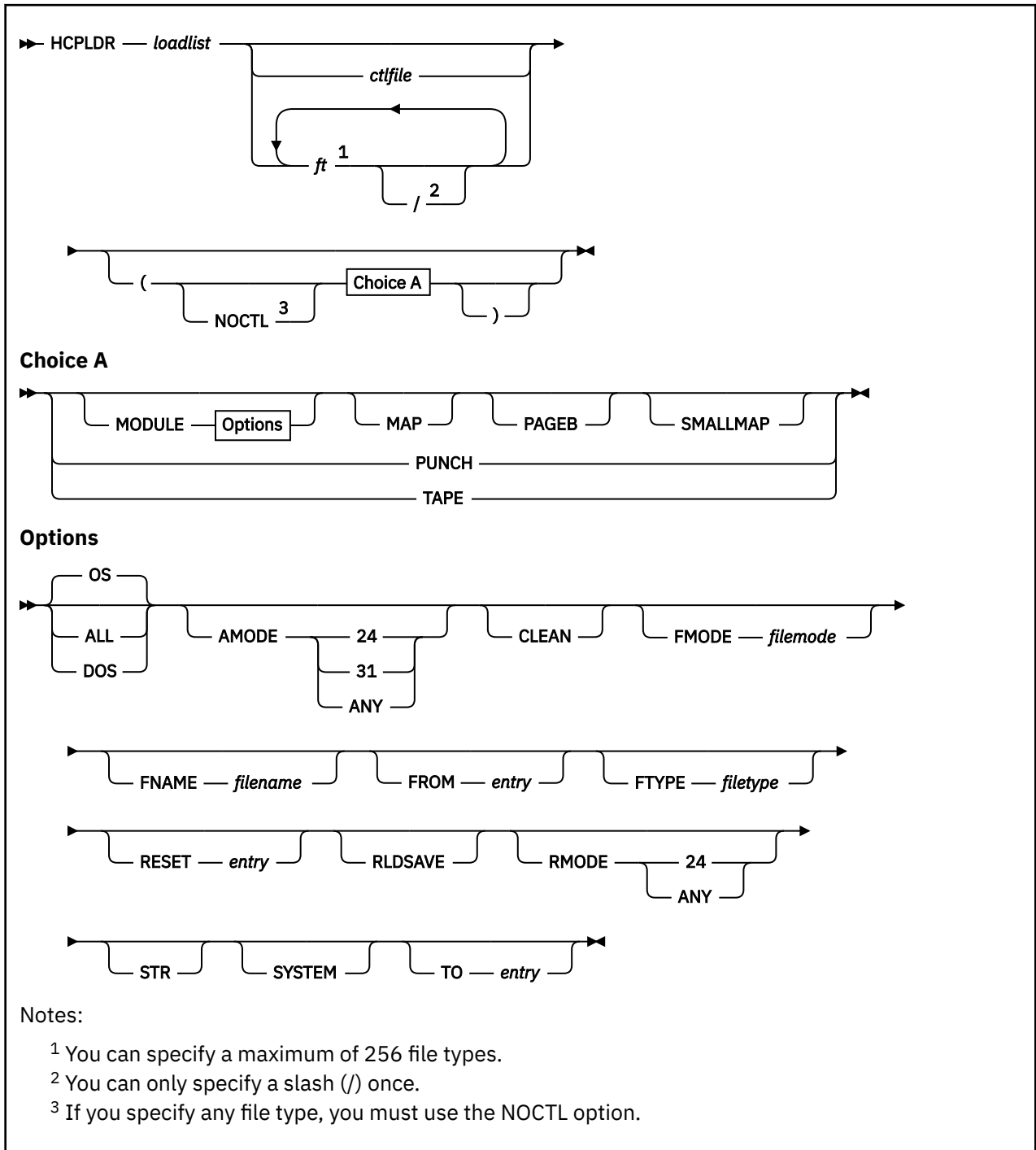
## **Usage Notes**

1. To produce image libraries, you must use the IMAGELIB utility to process the text decks that GENIMAGE creates.

## **Messages**

For information about the messages and return codes from this command, see the *IBM 3800 Printing Subsystem Programmer's Guide*.

# HCPLDR



## Purpose

Use the HCPLDR utility to call and control the system loader to perform one of two possible functions:

- If you enter the HCPLDR command without the PUNCH option or the TAPE option, the loader generates a system (directly under CMS). This means the system loader link-edits all object files and passes control to the program (HCPGEN) that writes the system on the system residence device.

- If you enter the HCPLDR command with either the TAPE option or the PUNCH option, the loader creates a continuous card deck beginning with a three-card loader, continuing with a stand-alone version of the system loader, and ending with the object files belonging to the system to be generated. When you enter HCPLDR with the TAPE or PUNCH option to generate CP, the loader does not link-edit object modules and does not pass control to HCPGEN to write the system on the system residence device.

Issue the HCPLDR command under CMS.



**Attention:** You should use the VMFBLD EXEC to build nuclei. If you use the HCPLDR command, it does not load any serviced text decks contained in the nucleus. The HCPLDR command does not support loading text decks by PTF number.

## Operands

### *loadlist*

is the file name of the load list EXEC file. It contains the names of the object files in the order in which they are to be loaded. The load list may look like this.

```
&CONTROL OFF
&1 &2 &3 HCPLDR LOADER
&1 &2 &3 fn [ft]
&1 &2 &3 fn [ft]
⋮
```

*fn* and *ft* are the file name and file type of an object file. The *ft* is optional. If *ft* is omitted, HCPLDR searches the control file or an explicit list of file types to determine the file type.

HCPLDR LOADER must always be the first file. HCPLDR loads each object module in order from the top of the load list to the bottom.

You can place most loader control statements in the load list EXEC file. You cannot place VERIFY, REPLACE, or INCLUDE CONTROL SECTION statements in the EXEC file. A loader control statement begins with a 12-2-9 punch (X'02'). Whenever HCPLDR encounters X'02' in the first column of a record, it interprets that record as a control statement. For more information see Appendix D in [z/VM: CP Commands and Utilities Reference](#).

### *ctlfile*

is the file name of the control file.

HCPLDR uses the update level identifiers in the control file to determine the file type for those object modules listed in the load list without file types. The command skips the MACS record in the control file and searches for update level identifiers from the top of the control file down. If HCPLDR finds an update level identifier other than TEXT, the command appends the identifier to TXT to create a file type, then searches for the file (*fn* TXTxxxx). If HCPLDR cannot find an object file that has the first update level identifier, the command finds the next identifier and continues the search. As a last resort, when HCPLDR cannot find the object file according to update level identifiers, the command searches for *fn* TEXT. For example, if the control file is:

```
TEXT MACS HCPGPI HCPSI HCPOM1 HCPOM2 DMSGPI DMSOM OSMACRO
SPEC AUX1111
IBM1 AUXVM
```

HCPLDR follows this search order:

```
fn TTXSPEC
fn TTXIBM1
fn TEXT (last resort)
```

If an asterisk (\*) or a period appears in the first column of a control file record, HCPLDR ignores the record. The command also ignores update level identifiers of PTF.

### *ft* [./] [...*ft*]

is a list of file types (up to 256) that you want HCPLDR to use in lieu of a control file. HCPLDR establishes a search order from left to right. The list of file types can be null, in which case a file type must be specified for each object module in the load list EXEC.

If you place a slash (/) anywhere in the file type list, HCPLDR notifies you when it loads an object module that has the same file type as the one that is to the left of the slash.

You must use the NOCTL option if you specify file types directly on the HCPLDR command line.

For example, assume you want to establish the following search order for HCPLDR:

```
fn TXT1
fn TXT2
fn TXT3
```

You also want to be notified when HCPLDR uses file type TXT1 or TXT2. You are using the load list CPLOAD EXEC. You would enter the following command:

```
hcpldr cpload txt1 txt2 / txt3 (noctl
```

When you enter a list of file types on the command line, HCPLDR does not automatically look for *fn* TEXT.

## Options

### NOCTL

allows you to specify a list of file types to be used in the search in place of using a control file.

### MODULE

requests that the object files be linked together as a module and be written to disk. The file name of the module will default to the name of the loadlist, the file type will default to MODULE and the file mode will default to A. The default file name, file type, and file mode may be overridden by using the FNAME, FTYPE, and FMODE options.

### MAP

requests the production of a load map during the loading process. This map is written in the file *loadlist* MAP A1. It is a variable length file with ASA carriage control and a maximum record length of 133 characters.

### PAGEB

requests the unconditional page boundary be initialized to 4 KB. The lower three hexadecimal digits of the address are the same as the assembled address for any single CSECT assembly. Use PAGEB only for systems that you are testing.

### SMALLMAP

requests the production of a smaller version of the standard load map (MAP option) during the loading process. The smallmap is written in the file *loadlist* SMALLMAP A1. It is a variable-length file and contains one line for each csect that was included. Both MAP and SMALLMAP can be specified at the same time. SMALLMAP cannot be used with the PUNCH or TAPE option.

### PUNCH

requests the object files be punched on the punch at virtual address 00D. A CP CLOSE command is issued when all of the object files have been punched.

### TAPE

requests the output of object files be placed on a tape drive at virtual address 182.

### OS

indicates the program may contain OS macros and therefore should be run only when CMS/DOS is not active. OS is the default.

### ALL

indicates that one or more of the following are true about the program:

- It contains CMS macros and must be capable of running regardless of whether CMS/DOS is active
- It contains no VSE or OS macros
- It preserves the value of the DOS flag on entry into the CMS nucleus, and, if it modifies that value, resets it on exit
- It does its own setting of the DOS flags.

**Note:** The ALL option is primarily for use by CMS system programmers. CMS system routines know which environment is active and preserve and reset the DOS flag in the CMS nucleus.

**DOS**

indicates the program contains VSE macros; CMS/DOS must be active (that is, SET DOS ON must have been previously invoked) for this program to run.

**AMODE**

specifies the addressing mode in which the program will be entered. The AMODE defined by this option is placed in the header record of the MODULE file being created.

If you specify RMODE but do not specify AMODE, the AMODE for the MODULE is determined from the following default criteria:

- If you specify RMODE ANY, the AMODE specification defaults to AMODE 31.
- If you specify RMODE 24, the AMODE defaults to the AMODE of the entry point determined by the LOAD process.

If you specify neither AMODE nor RMODE, the AMODE for the module defaults to 31.

The valid AMODE values and their meanings are as follows:

**24**

This entry point is to receive control in 24-bit addressing mode.

**31**

This entry point is to receive control in 31-bit addressing mode.

**ANY**

tells HCPLDR that the specified entry point is capable of operating in either 24-bit or 31-bit addressing mode. It will receive control in the addressing mode of its caller. If the module is run by name from the command line or from an exec, it will receive control in 31-bit addressing mode.

**CLEAN**

tells CMS to erase the module from storage at the end of its execution. This option will work only for modules running under CMS. CLEAN is the default for relocatable modules.

**Note:** You cannot specify the CLEAN option for nonrelocatable modules.

**FMODE *filemode***

specifies the file mode of the module being generated. If omitted, *filemode* defaults to A.

**FNAME *filename***

specifies the file name of the module being generated. If omitted, *filename* defaults to the name of the load list.

**FROM *entry***

specifies an entry point or a control section name that represents the starting virtual storage location from which a nonrelocatable MODULE is generated.

**FTYPE *filetype***

specifies the file type of the module being generated. If omitted, *filetype* defaults to MODULE.

**RESET *entry***

sets the starting location for the module. The operand *entry* must be an external name (for example, CSECT control section or ENTRY) in the module. If RESET is not specified, the default entry point is used.

**Note:** You should not use the RESET option when creating a module from text files created by any of the following OS/VS language processors under CMS:

- OS Code and Go FORTRAN
- OS FORTRAN IV (G1)
- OS FORTRAN IV (H) Extended
- OS/VS COBOL Compiler and Library
- OS Full American National Standard COBOL Version 4 Compiler and Library.

**RLDSAVE**

instructs HCPLDR to save the relocation information from the text files and to create a relocatable module.

**RMODE**

specifies the location in virtual machine storage where the loaded MODULE is to reside.

If you specify AMODE but do not specify RMODE, the RMODE for the module defaults to 24.

If you specify neither AMODE nor RMODE, the RMODE for the module defaults to ANY.

The valid RMODE values and their meanings are as follows:

**24**

The load must reside below the 16 MB line, overriding the RMODE determined by the LOAD process. A definition of RMODE 24 would be reflected in the MODULE header record.

**ANY**

The load may reside above or below the 16 MB line, overriding the RMODE determined by the LOAD process. A definition of RMODE ANY would be reflected in the MODULE header record.

**STR**

deletes previously loaded non-OS programs when used in conjunction with the NOPRES option on the CMS LOADMOD command.

**SYSTEM**

indicates that when the MODULE file is subsequently loaded, it is to have a storage protect key of zero.

**TO entry**

specifies an entry point or a control section name that represents the ending virtual storage location from which a nonrelocatable MODULE is generated.

**Usage Notes**

1. When you use the system loader as a system generator, your virtual machine must have sufficient storage to contain CMS **plus** the entire system you are generating. If the system is too large to generate directly, use the stand-alone version of the system loader.
2. Using the PAGEB option improves the debugging task because it eliminates needless arithmetic (the lower three hexadecimal digits of the load address is the same as the assemble address). Note, however, that PAGEB should only be used for testing purposes because the option wastes storage.
3. After the new nucleus or module resides on the system residence device, you can shut down your present system (with the SHUTDOWN command) and perform a hardware load of the new system.
4. The following response will appear if you create the CP module with the MAP option but without the PAGEB option:

```
CSECT'S WITH SIZE GREATER THAN CONDITIONAL PAGE BOUNDARY
@MAPSTRT xxxxxxx
```

This response is for information only and is no cause for concern unless other CSECTs are listed. If other CSECTs are listed, notify the IBM Support Center.

5. Although the RLDSAVE operand is accepted when generating the CP module, it has no effect on CP. CP will always relocate itself back to X'2000'.

**Messages**

- HCP003E Invalid option - {*option*|command contains extra option(s) starting with *option*}
- HCP013E Conflicting option - *option*
- HCP038E Parameter for *option* option missing or invalid
- HCP863E Invalid parameter - *parameter*
- HCP6878E AMODE/RMODE VALUES CONFLICT
- HCP6879E THE LENGTH OF THE MODULE TO BE GENERATED IS NON-POSITIVE

## HCPLDR

- HCP8000E FILE *fn ft fm* NOT FOUND
- HCP8001E FILE *fn ft fm* IS NOT FIXED, 80 CHAR. RECORDS
- HCP8003E DATA NOT REPLACED: *text*
- HCP8005E DUPLICATE CSECT - *csectname*
- HCP8006E *fn ft fm* DOES NOT HAVE FIXED, 80-BYTE RECORDS
- HCP8007E UNEXPECTED ERROR *nnn* - PLIST *plist fn ft fm*
- HCP8008E PUNCH NOT ATTACHED
- HCP8009E ERROR ON PUNCH
- HCP8010W NO STARTING ADDRESS
- HCP8011E VERIFY FAILURE: *text*
- HCP8012E ALREADY DEFINED CSECT: *text*
- HCP8013I POSSIBLE OVERLAY: *text*
- HCP8014E DUPLICATE ENTRY: *text*
- HCP8015E INVALID CARD FORMAT: *text*
- HCP8016E INVALID CARD: *text*
- HCP8017E INVALID DATE: *text*
- HCP8018E INVALID FORMAT: *text*
- HCP8019E INVALID PLACEMENT: *text*
- HCP8020E NO ARGUMENT SPECIFIED: *text*
- HCP8021E OPERAND MISSING: *text*
- HCP8022E UNDEFINED ENTRY: *text*
- HCP8023E UNRECOGNIZED CARD *text*
- HCP8024E NO FILETYPE SPECIFIED FOR *filename*
- HCP8025E NO CSECTS DEFINED
- HCP8026E ERROR *nnn* WRITING TO TAPE
- HCP8027W PERMANENT ERROR ON INPUT
- HCP8028W INSUFFICIENT STORAGE AVAILABLE
- HCP8029E DUPLICATE ESDID DEFINITION; MISSING 'END' CARD
- HCP8030E NO OBJECT FILE FOUND FOR *filename*
- HCP8031W PROGRAM INTERRUPT DURING LOADING OPERATION
- HCP8032W ATTEMPT TO LOAD FAULTY STORAGE
- HCP8033E NO LOAD LIST SPECIFIED
- HCP8034E NO CONTROL FILE SPECIFIED
- HCP8035E DUPLICATE FILETYPE SPECIFIED - *filetype*
- HCP8036E TOO MANY FILETYPES SPECIFIED
- HCP8037W A PROGRAM CHECK OCCURRED IN HCPLDR WHILE RELOCATING THE TARGET PROGRAM.
- HCP8038W A MACHINE CHECK OCCURRED IN HCPLDR WHILE RELOCATING THE TARGET PROGRAM.
- HCP8039S ERROR *nnn* WRITING FILE *fn ft fm* ON DISK
- HCP8049E ERROR DURING ERASE OF *fn ft fm*. FSERASE RC=*nnn*
- HCP8049I ERROR DURING ERASE OF *fn ft fm*. FSERASE RC=*nnn*



# IMAGELIB

▶ IMAGELIB — *libname* ◀

## Purpose

Use the IMAGELIB utility to create or replace an image library with images in text decks. The output image library can then be referenced by a printer by issuing the START command with the IMAGE option for the printer.

## Operands

### *libname*

identifies in one to eight characters the image library that you want to create or replace. This name is also the name of the new or replacement library. The control file that lists the members that you want to create or replace also must have this name as its file name (see Usage Note “2.b” on page 2201).

## Usage Notes

1. The IMAGELIB utility can invoke one or more CP commands that require a user privilege class other than G.
2. Before you enter the IMAGELIB command, you must do two things:
  - a. First, follow the procedure that applies to your type of printer.
    - For 3800 images, enter the GENIMAGE utility to create the text decks that contain the images for the new library.
    - For impact printer images, code the appropriate FCB or UCS macro in an ASSEMBLE file. Assemble this file to create the TEXT deck for each image.
  - b. Create a control file whose file name is the same as the IMAGELIB you want to build, and whose filetype is CNTRL. The format of this file is one statement per TEXT deck you want to include, with the names of the decks in columns 1 through 8.
3. You may use the IMAGELIB utility with these printers:
  - 1403
  - 3203
  - 3262
  - 3800
  - 4245
4. Using VMRELOCATE for a virtual machine that might be using this function is not recommended and may have undesired results. If a virtual machine reads an IMAGELIB and is then relocated to another system, any subsequent save of that IMAGELIB will be written out to a different system, not the one from which it was read.

## Messages

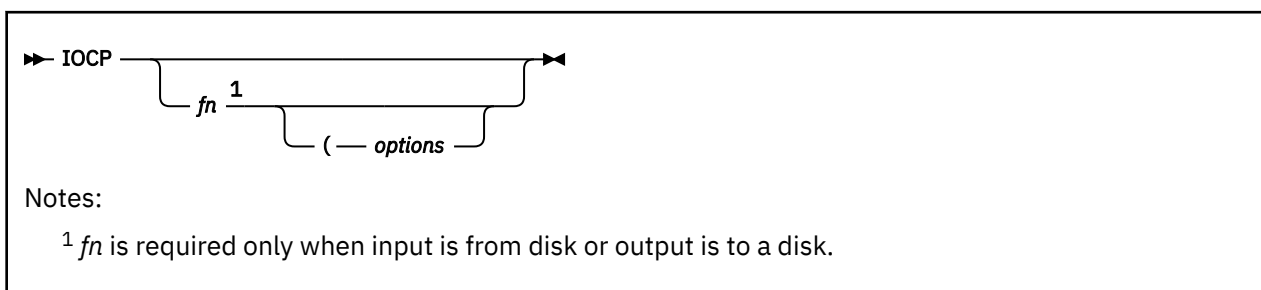
- HCP247I NAMED IMAGE *imagename* CREATED
- HCP248E SPECIFIED IMAGE *imagename* NON-EXISTENT
- HCP249E ERROR LOADING IMAGE *imagename*
- HCP253E RESIDUAL BYTE COUNT = *nnnnnnnn* (HEX)
- HCP254E ERROR SAVING *imagelib*

## IMAGELIB

- HCP759E INSUFFICIENT VIRTUAL STORAGE

**Note:** If the control file discussed in Usage Note [“2.b” on page 2201](#) does not exist or is not valid, IMAGELIB issues a return code of 4 in register 15, but does not enter an error message.

## IOCP



### Purpose

Use the IOCP utility to call the input/output configuration program for the following reasons:

- To generate a new input/output configuration data set (IOCDS) from a file containing source IOCP statement
- To produce configuration reports from an existing IOCDS

### Operands

#### *fn*

is the name of the file that contains the source IOCP statement or of the output listing file, depending on the options you specify. *filetype* must be IOCP.

#### *options*

are the options passed to the IOCP utility. The IOCP utility consists of a front-end IOCP exec that examines and possibly alters the IOCP input file before giving control to the IOCP program. The options that are listed here are only used and interpreted by the front-end IOCP exec. They are not passed on to the IOCP program. The options for the IOCP program are not described here. For a complete description of the IOCP program options, see the *IBM Z: Input/Output Configuration Program User's Guide for ICP IOCP*.

#### **DYN**

is an outdated option. Specifying DYN or NODYN no longer has any effect on the operation. Doing so only triggers a warning message. The issuer's privilege class now determines whether dynamic changes are allowed to their I/O configuration, not DYN and NODYN. See Usage Note "3" on page 2204 for more details.

#### **NODYN**

is an outdated option. See the description of the DYN parameter for more details.

#### **DESC1 string**

indicates that you want the specified string to be stored in the first descriptor field of the hardware configuration token. The string can be up to eight characters and cannot contain blanks.

#### **DESC2 string**

indicates that you want the specified string to be stored in the second descriptor field of the hardware configuration token. The string can be up to eight characters and cannot contain blanks.

### Usage Notes

1. For more information about using the IOCP utility and making dynamic changes to your I/O configuration, see *z/VM: CP Planning and Administration*. For additional information about the IOCP program, see *IBM Z: Input/Output Configuration Program User's Guide for ICP IOCP*.
2. If the IOCP front-end exec terminates without invoking the IOCP program, an error return code of the format (1000+nnn) is generated. The nnn value provides additional information about the problem that was encountered:

**nnn****Description****sss**

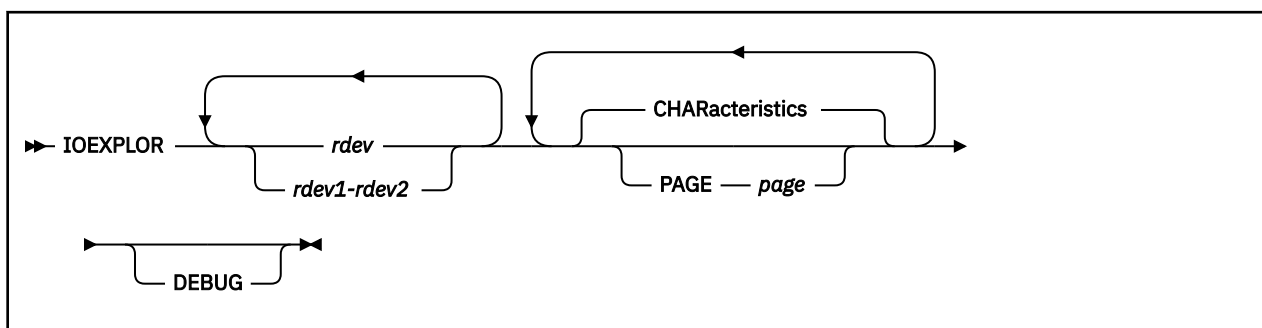
Changes to the IOCP file could not be saved. This value is the error return code received from the XEDIT SAVE command. See *z/VM: Enterprise Systems Architecture/Extended Configuration Principles of Operation* for additional information as to the nature of the problem.

The IOCP program will also generate return codes. For details on the return codes, see *IBM Z: Input/Output Configuration Program User's Guide for ICP IOCP*.

3. If you want to create an IOCDs that will allow dynamic changes to your I/O configuration, you must have privilege class B so that the CP QUERY TOKEN command can be used. The front-end IOCP exec examines and changes the IOCP file to ensure a valid hardware configuration token is present in the IOCP file before invoking the IOCP program. The IOCP file must be on a read/write disk so that the change to the IOCP file can be saved. If the change cannot be saved, the front-end IOCP exec will terminate without invoking the IOCP program.

If you do not have the correct privilege class to make dynamic changes to your I/O configuration, the IOCP file will be examined; and if the IOCP file contains a hardware configuration token, it will be removed before invoking the IOCP program. If the IOCP file contains a hardware configuration token that must be removed, then the IOCP file must be on a read/write disk so that the change can be saved. If the change cannot be saved, the front-end IOCP exec will terminate without invoking the IOCP program.

## IOEXPLOR



### Purpose

Use the IOEXPLOR utility to obtain device information. For IBM devices, this utility formats device characteristics obtained from the CP QUERY EDEVICE and QUERY DASD commands. For non-IBM devices, data which is not vendor specific will be formatted.

### Operands

**rdev**

**rdev1-rdev2**

Specifies the devices for which you want information to be displayed. A device in the active configuration can be entered as a 4-digit hexadecimal device number between X'0000' and X'FFFF'. For a device in the active or standby configuration, the device can be a 5-digit hexadecimal number between X'00000' and X'3FFFF' with the leading digit specifying the subchannel set. A range of devices cannot span subchannel sets.

### Options

#### CHARacteristics

Returns characteristic data for a device.

- For a connected EDEVICE, this data is selected from standard inquiry data, unit serial number page (80), device identification page (83), and Read Device Characteristics data. For IBM devices, the data is formatted to the console. For non-IBM devices, data which is not vendor specific will be formatted to the console. Control will be given to a vendor format routine if one exists.
- For a FICON-connected device, output from Read Device Characteristics and Read Configuration Data is formatted to the console.

#### PAGE *page*

Use the PAGE option to specify a vital product data page (*page*) to have that data returned for a connected emulated device. The *page* value must be a hexadecimal number between X'00' and X'FF'. The data returned is a hexadecimal data block unless a format routine exists. Format routines can be written for each device manufacturer, one per manufacturer, to format the output from the QUERY EDEVICE command for specific vital product data pages.

#### DEBUG

displays general and diagnostic information during program execution. Results of this option are subject to change.

### Format Routine

IOEXPLOR calls a format routine if the PAGE option is specified. This routine could format the hexadecimal output from the QUERY EDEVICE command with the INQUIRE option specified. The routine has a file name that consists of the manufacturer's name concatenated with "DEVF". If the name consists of more than four characters, only the first four characters will be used. The file type must be EXEC.

For IBM devices, the data from the serial number page (80) and the device identification page (83) is formatted by the IOEXPLOR utility.

IOEXPLOR will call the routine with the specified page number and the hexadecimal data returned from the QUERY EDEVICE command. The output is obtained from the data returned by the SCSI INQUIRY command. Parts of the response are vendor-specific and could vary from one vendor to another. Consult your vendor documentation if you plan to supply a custom formatting routine. IOEXPLOR will format the data that is not vendor specific.

It is expected that the routine will format the data as desired and display it on the console.

## Usage Notes

1. This utility issues the QUERY EDEVICE command and the QUERY DASD command, which require class B privilege.
2. The maximum length of the inquiry response data display is 255 bytes.
3. When the CHARACTERISTICS option and the PAGE option are both specified, the PAGE option is ignored for FICON-connected devices.
4. When a range of devices is specified, information about each device is displayed.
  - a. All emulated devices will have the requested CHARACTERISTICS data and PAGE information. If a device in the range does not support the requested page, an error message will be displayed for that device.
  - b. All FICON-connected devices will have the requested CHARACTERISTICS data returned. If the PAGE option was specified, it will be ignored. If only PAGE data was requested, an error message will be displayed for each device.
  - c. If a device in the range does not exist, an error message is displayed for that device.

## Responses

Response 1:

The following is a flash system example.

```
IOEXPLOR 0125 or IOEXPLOR 0125 CHAR

-Begin: Characteristics Data for device 0125
Serial Number _____ 052a62e2052a-0000-0027-000029
      Standard Inquiry Data
Peripheral Qualifier/Peripheral Device Type __ 000b/00h
Vendor Identification _____ IBM
Product Identification _____ FlashSystem-9840
Product Revision Level _____ 1217
Version Descriptor _____ SAM-3 (no version)
Version Descriptor _____ FC-PH-3 (no version)
Version Descriptor _____ FC-AL-2 (no version)
Version Descriptor _____ FCP-3 (no version)
Version Descriptor _____ SPC-3 (no version)
Version Descriptor _____ SBC-2 (no version)
      Device Identification
Cluster Identification _____ 052a62e2052a
IO Group _____ 0000
Vdisk Number _____ 0027
LUN Identification _____ 000029
IEEE Company Identification _____ 005076
Cluster Alias _____ 12A62E2052A
Slot Number _____ 10
Channel Number _____ 01
Extent Space Efficient Supported _____ Yes
      Device Characteristics
Device class code _____ 21
Unit type _____ 11
Bytes per track _____ 56832
Bytes per cylinder _____ 397824
Bytes per block _____ 512
Device size _____ 2097152 blocks
-End: Characteristics Data for device 0125
```

## Response 2:

The following is an example of CHARACTERISTICS data for a DS8K device.

```

IOEXPLOR 7504 or IOEXPLOR 7504 CHAR

-Begin: Characteristics Data for device 7504
Serial Number ----- 75041911015
      Standard Inquiry Data
Peripheral Qualifier/Peripheral Device Type ----- 000b/00h
Vendor Identification ----- IBM
Product Type-Model ----- 2107-900
Microcode EC Level ----- .204
Plant of Manufacture ----- 75
Storage Facility Sequence Number ----- 04191
Unique Identifier for LUN ----- 1015
Version Descriptor ----- SAM-3 (no version)
Version Descriptor ----- FC-FS (no version)
Version Descriptor ----- FCP-3 (no version)
Version Descriptor ----- SPC-3 (no version)
Version Descriptor ----- SBC-2 (no version)
      Device Identification
IEEE Company Identification ----- 05076
Vendor Specified ----- 303FFC09C
LUN Identifier (LSS.Vol) ----- 10.15
Relative Target Port ----- 00000102
Target Port Group Identifier ----- 00000000
Extent Space Efficient Supported ----- Yes
      Device Characteristics
Device class code ----- 21
Unit type ----- 11
Bytes per track ----- 56832
Bytes per cylinder ----- 397824
Bytes per block ----- 512
Device size ----- 2097152 blocks
-End: Characteristics Data for device 7504

```

## Response 3:

The following is an example of CHARACTERISTICS data for a XIV device.

```

IOEXPLOR 0500 or IOEXPLOR 0500 CHAR

-Begin: Characteristics Data for device 500
Serial Number ----- 78030BB03A7
      Standard Inquiry Data
Peripheral Qualifier/Peripheral Device Type ----- 000b/00h
Vendor Identification ----- IBM
Product Type-Model ----- 2810-XIV
Microcode EC Level ----- 0000
Machine Serial Number ----- 33304242
Volume Number ----- 30334137
      Device Identification
Extended Unique Identifier ----- 0017380030BB03A7
Company Identification ----- IBM
Model Type ----- 2810XIV
Plant of Manufacture ----- 78
Machine's Serial Number ----- 030BB
Volume Identification ----- 03A7
Volume Name ----- s34_all_LPARs
Extent Space Efficient Supported ----- Yes
      Device Characteristics
Device class code ----- 21
Unit type ----- 11
Bytes per track ----- 56832
Bytes per cylinder ----- 397824
Bytes per block ----- 512
Device size ----- 33609728 blocks
-End: Characteristics Data for device 500

```

## Response 4:

The following is an example of a FICON-connected DASD.

```

IOEXPLOR 550A

```

```

-Begin: Characteristics Data for device 550A
      I/O Device Information
Device type-model ----- 2107-900
Device manufacturer ----- IBM
Serial number (plant-seq#) ----- 75-YC321
Logical Volume Number ----- 450A
      Control Unit Information
Device type-model ----- 2107-961
Serial number (plant-seq#) ----- 75-YC321
Logical Subsystem Number ----- 45
      Additional Device Information
Device manufacturer ----- IBM
Device type-model ----- 2107-961
Serial number (plant-seq#) ----- 75-YC320
Logical Subsystem Number ----- 45
      Additional Device Information
Device manufacturer ----- IBM
Device type-model ----- 2107-900
Serial number (plant-seq#) ----- 75-YC321
Logical Subsystem Number ----- 45
Interface id ----- 0530
Missing Interrupt Timer Interval ----- 30 seconds
Secondary Missing Interrupt Timer Interval ----- 0 seconds
Controller System Adapter ID (SAID) ----- 0530
Logical paths supported ----- 81920
      Device
Host CU type-model ----- 2107-E8
Device type-model ----- 3390-0A
      Storage Directory Facilities
VM non-full pack minidisk ----- No
MIDAW Capability supported ----- No
Parallel Access Vol. state ----- HyperPAV Enabled
XRC Functions ----- Enabled
Peer-to-Peer Remote Copy ----- Enabled
Striping and Compaction ----- Supported
Locate Record Erase ----- Supported
Cache Fast Write ----- Supported
Multi-Path Lock ----- Supported
Track Cache ----- Supported
DASD Fast Write ----- Supported
24 Byte Compatibility sense ----- Yes
Device class code ----- 20
Device type code ----- 24
Primary cylinders ----- 3339
Tracks per cylinder ----- 15
Number of Sectors ----- 224
Track length ----- 58786
HA + R0 length ----- 1428
Capacity formula ----- 2
Capacity factors F1-F6 ----- 34 19 9 6 116 6
MDR Record ID ----- 24
OBR Record ID ----- 24
Storage director Type ----- 1F
Read Trackset length ----- 2
Max Record zero length ----- 57326
      Storage Class
Data Encrypted device ----- No
Solid State drive ----- No
Enterprise Disk ----- Yes
SATA Disk ----- No
Flash Storage ----- No
Tier 1 or 2 Flash ----- No
Tiered Storage Pool ----- No
Track Set Size ----- 1
RPS Sector Calc. Factors F7-F8 ----- 77 08
Concurrent Copy Lower ----- 0F
Concurrent Copy Upper ----- 7F
      Generic Device / CU functions
Mirrored Device ----- No
RAID Device ----- Yes
Transparent subsystem cache ----- No
Split CE/DE ----- Yes
Device capable of Mirroring ----- No
XRC Device Management enabled ----- Yes
RVA Snapshot supported ----- No
Real Control Unit code ----- 00
Real Device Code ----- 3C
Extent Space Efficient Supported ----- No
Storage class ----- 10K/15K RPM HDD
-End: Characteristics Data for device 550A

```



## Response 5:

The following is an example of using the PAGE option with an XIV device. PAGE 83 is formatted by IOEXPLOR.

```
IOEXPLOR 0500 PAGE 83

-Begin: Characteristics Data for device 500
      Device Identification
Peripheral Qualifier/Peripheral Device Type _____ 000b/00h
Extended Unique Identifier _____ 0017380030BB03A7
Company Identification _____ IBM
Model Type _____ 2810XIV
Plant of Manufacture _____ 78
Machine's Serial Number _____ 030BB
Volume Identification _____ 03A7
-End: Characteristics Data for device 500
```

## Response 6:

The following is an example of using the PAGE option with a Flash device. In this case, page 00 is not formatted by IOEXPLOR because it is not one of the supported pages. The hexadecimal data from the QUERY EDEVICE command is displayed.

```
IOEXPLOR 0125 PAGE 00

- Begin - EDEV 0125 - Vital Product Data Page 00 -
00000006 00808386 B0B1
- End - EDEV 0125 - Vital Product Data Page 00 -
```

## Response 7:

Some variations in the returned information, where 019A is a FICON-connected device.

## 1. IOEXPLOR 019A PAGE 00

Returns: HCP972I The PAGE option is not valid for device 19A

## 2. IOEXPLOR 019A CHAR PAGE 00

Returns the CHARACTERISTICS data for 019A and ignores the PAGE option.

## Response 8:

Some variations in the returned information, where 019A is a FICON-connected device, 019B is an emulated device, and 019C does not exist.

## 1. IOEXPLOR 019A-019C CHAR

- Returns CHARACTERISTICS data for 19A and 19B
- Returns: HCP040E Device 019C does not exist

## 2. IOEXPLOR 019A-19C CHAR PAGE 00

Returns similar results as the previous example and the emulated device 019B would also display the hex data for page 00.

## Messages

- HCP040E Device *rdev* does not exist
- HCP970E Invalid range specified
- HCP971E Missing operand(s)
- HCP972I The PAGE option is not valid for device *rdev*
- HCP974E *option* is not a valid command line option

## IPWIZARD

---

▶ IPWIZARD ◀

### Purpose

Use the IPWIZARD utility to create a minimal TCP/IP configuration that establishes basic connectivity to your IP network. The command displays a panel requesting network information. After you fill out the panel, the information is processed and the TCP/IP SYSTEM DTCPARMS, TCPIP DATA, and PROFILE TCPIP files are created.

### Usage Notes

1. IPWIZARD requires access to the MAINT 193 and 2CC minidisks. This utility is intended to be used only as part of a z/VM system installation procedure.

### Messages

- HCP8300E FILE *filename filetype* NOT FOUND
- HCP8312E ERROR DISPLAYING HELPFIELD *name*
- HCP8330E DEVICE ADDRESS MUST BE BETWEEN *addr1* AND *addr2*
- HCP8342E THE COMMAND *command* FAILED WITH RC=*rc*
- HCP8352E INVALID MTU SIZE ENTERED
- HCP8353W UNDEFINED PFKEY
- HCP8357I INVALID IP ADDRESS ENTERED
- HCP8376E IPWIZARD EXEC ENDED IN ERROR
- HCP8392I IPWIZARD EXEC ENDED SUCCESSFULLY
- HCP8401E IPWIZARD EXEC MUST BE RUN ON A FULL SCREEN TERMINAL {WITH AT LEAST 20 LINES | WITH AT LEAST 80 COLUMNS}
- HCP8431E THE *addr* DISK MUST BE ACCESSED AS *filemode* IN R/W MODE TO RUN IPWIZARD
- HCP8471W ONLY ONE *selection* MAY BE SELECTED
- HCP8472I YOU MUST { SELECT | INPUT } *selection* BEFORE PRESSING PF*n*

Refer to [IPWIZARD Messages](#) in *z/VM: TCP/IP Messages and Codes* for information about the DTCIPW messages you may receive.

## MIGR51D

►► MIGR51D ◄◄

### Purpose

Use the MIGR51D utility to update the System Software Inventory files for this z/VM release from the inventory files of your previous z/VM release. MIGR51D displays panels that allow you to select which products and segments to migrate and not to migrate.

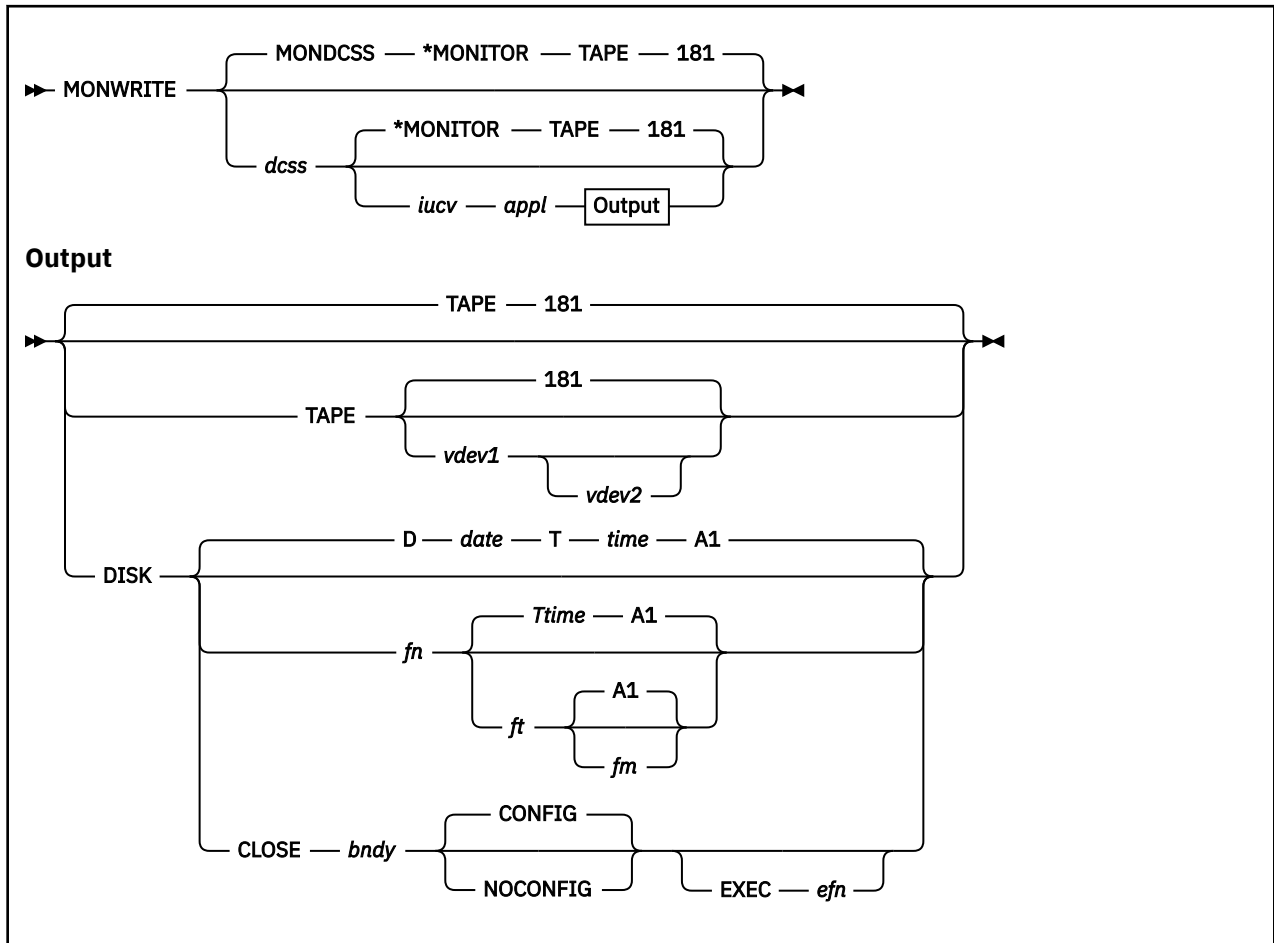
### Usage Notes

1. By default, the MIGR51D utility is located on the system tools disk (MAINT 193). MIGR51D requires write access to the MAINTvrm 51D minidisk. This utility is intended to be used only as part of a z/VM system installation procedure.
2. The MIGR51D utility logs issued messages and commentary about its principal actions (file copy, rename, or updates) in a MIGR51D MSGLOG file.
3. If errors arise during MIGR51D program execution, review the MIGR51D MSGLOG file for detailed information about files that might need to be renamed to allow for a successive MIGR51D command to complete successfully.

### Messages

- IUG8300E File *filename filetype filemode* [( *userid vdev* ) | (*dirID*)] not found
- IUG8312E Error displaying HELP file *filename*
- IUG8342E Command *cmdname* failed with RC=*rc* [ while processing file *file\_ID* ( *device info* ) ]  
[ Command arguments: *data* ]
- IUG8353W Undefined function key
- IUG8385W Cannot migrate segment *segmentname*. Segment name must be changed
- IUG8401E *cmdname* must be run on a full screen terminal with at least *nn* {lines|columns}. [ Your terminal contains *nn* lines. ]
- IUG8431E The { *vdev disk* | *dir\_ID* directory } must be accessed { as filemode *filemode* | in *status* mode } to run *cmdname*
- IUG8431E Disk *vdev* must be linked
- IUG8431E A { *vdev disk* | *dir\_ID* directory } must be linked in write mode to run *cmdname*
- IUG8463E { *input* [specified] | Command entered } is not valid[{:|.}] [{{*input*|reason}}]
- IUG8477E A temporary MIGR51D file has been found on the {current|previous} release Software Inventory disk (51D). This disk must be restored prior to restarting MIGR51D.
- IUG8478R Enter the filemode letter of the Software Inventory disk (51D) from the previous release. Press 'Enter' to Exit.
- IUG8480E The Software Inventory disk for the previous release did not pass validity checking. Please correct and re-issue *cmdname*
- IUG8481I Exiting *cmdname* at user request
- IUG8499E The *table\_name* table contains the following duplicate key entries:
- IUG8990I *command* processing completed successfully (RC=*rc*)
- IUG8991W *command* processing completed with warnings (RC=*rc*)
- IUG8992E *command* processing completed unsuccessfully (RC=*rc*)

## MONWRITE



### Purpose

Use the MONWRITE utility to retrieve monitor records from the monitor saved segment and store them in a CMS file or on a tape.

To stop the MONWRITE utility, use the MONWSTOP command. See “MONWSTOP” on page 2216 for more information.

### Operands

#### *dcss*

identifies the saved segment defined and saved for monitor. MONDCSS is the default name.

#### *iucv*

identifies the IUCV server that notifies the monitor writer about data in the saved segment. \*MONITOR is the default server.

#### *appl*

identifies the IUCV application program running in the virtual machine specified by *iucv*. This option is required if you specify *iucv* and is not allowed if you specify \*MONITOR.

#### TAPE

specifies that the output will be written on a tape. TAPE is the default.

#### *vdev1*

specifies the virtual device number of the first tape on which the output will be written. The default virtual device number is 181.

If *vdev2* is not specified (that is, only one tape is to be used), and *vdev1* becomes full, it rewinds and unloads from the tape drive and MONWRITE waits for a new tape to be mounted.

**vdev2**

specifies the virtual device number of the second tape on which the output will be written when the first tape, *vdev1*, becomes full. When *vdev2* becomes full, MONWRITE reuses the first tape, *vdev1*.

**DISK**

specifies that the output will be written to a CMS file.

**Ddate Ttime**

specifies the default date and time used for the file name and type of the monitor data file.

The format for the file name is *Dmmdyy*, where *mm* is the month (1-12), *dd* is the day (1-31), and *yy* is the last two digits of the year when the file was created.

The format for the file type is *Thhmmss*, where *hh* is the hour (00-23), *mm* is the minute (0-59), and *ss* is the second (0-59) when the file was created.

**fn ft fm**

identifies the file name, type, and mode of the CMS file to which the output will be written. The default is *Ddate Ttime A1*, where *date* and *time* are the date and time, respectively, when the file was created.

**CLOSE**

specifies that the output file containing the monitor records will be closed and a new file is created at interval *bdy*.

**bdy**

specifies an interval, in minutes, between 15 and 1440. Starting at midnight, at each such interval, the next time CP emits monitor sample records, MONWRITE closes the file and starts a new one.

**CONFIG**

specifies that each file created contains monitor configuration data. The connection to monitor is severed and reestablished upon each close and open.

**NOCONFIG**

specifies that only the first file created will contain monitor configuration data. The connection to monitor will not be severed between closing the old file and creating the new.

**EXEC**

indicates that a user exit EXEC will be executed.

When a user exit is invoked, the following parameters are passed:

- The file name and file type of the file closed just prior to the call of the user exit.

**efn**

specifies the user exit that will be executed. *efn* must be a CMS EXEC file with file name *efn* and file type EXEC. The default CMS file search order is used to determine where the EXEC file is located.

**Usage Notes**

1. To use the MONWRITE utility, the IUCV server (\*MONITOR, for example) must be specified on an IUCV statement in your directory entry.
2. MONWRITE connects to \*MONITOR in shared mode. It writes event data as long as it is active.
3. If other users are connected to \*MONITOR in shared mode, the name of the saved segment specified on the MONWRITE utility must be the same saved segment. (Issue QUERY MONITOR to determine the current saved segment.)
4. If another user is already connected in exclusive mode, the MONWRITE utility fails and a message is issued indicating that \*MONITOR severed the connection.
5. When the DISK operand is specified, the file name of the output file cannot be CLOSE.
6. The following considerations apply to the use of the CLOSE operand:
  - DISK must be specified. If TAPE is specified, an error message is issued and processing of the MONWRITE utility halts.

## MONWRITE

- The default date *Ddate* and time *Ttime* will be used for the file name and file type of the monitor data files.
  - The digits representing seconds in the time stamp, which make up the file type, are not significant and could vary from 0 to 59.
  - When a time zone change occurs, the current file is closed and a new file is opened. The time stamp of the new file is set according to the new time zone.
  - The *bdy* interval you specify is honored at only a moment when CP emits sample records. For example, suppose you enter CP MONITOR START at 15:35 and you specify a monitor sample interval of 30 minutes. This means monitor sample records arrive at *hh:35* and *hh:05*. Now suppose you enter MONWRITE CLOSE 60. Because the monitor sample interval occurs at *hh:05*, the MONWRITE file will be closed at *hh:05*.
7. The format of the MONWRITE output is described in *z/VM: Performance*; see [The MONWRITE Program](#) for more information.
  8. For tape output devices, MONWRITE uses the mode setting that exists for the device before you start the command. To obtain the proper mode setting, enter the TAPE MODESET command, specifying the TRACK and DENSITY parameters you want, for each device MONWRITE uses before using the MONWRITE utility.

When writing across multiple tape reels, it is possible for MONWRITE to miss data as it waits for a new tape to be mounted. It is recommended that two tape drives be mounted so that MONWRITE does not lose any data.
  9. CMS recognizes symbolic names for certain virtual devices, so you can use a symbolic name as the virtual device. For example, TAP1 is equivalent to 181.

### Examples

1. To collect data and send it to tape 181, enter:

```
monwrite
```

**Note:** This form of the command provides the easiest operation for the IBM 3480 tape drive with the cartridge stacker loader, which handles end-of-tape conditions automatically.

2. To collect data and send it to a disk file with a file ID that contains the date and time, enter:

```
monwrite mondcss *monitor disk
```

3. To collect data on tape using addresses 181 and 185, enter:

```
monwrite mondcss *monitor tape 181 185
```

4. In this example, you want to collect data from a monitor DCSS called CPMON.

- The IUCV server is running in the USERVM virtual machine.
- The name of the application program, as specified in the HNDIUCV SET macro, is WALKMON.
- Output goes to the file ID SPECIAL MONFILE C0. If this file already exists, MONWRITE appends the data to the end of the file.

Be sure to access your C-disk, then enter:

```
monwrite cpmo uservm walkmon disk special monfile c0
```

### Messages

- HCP002E Invalid operand - *operand*
- HCP026E Operand missing or invalid
- HCP2400E User exit EXEC *execname* not found
- HCP2401I Unexpected error from EXEC *execname* rc *code*

- HCP6265A A Monitor writer connected to *iucvserver action*
- HCP6266A End-of-tape encountered for *vdev*, mount and ready a new writable tape
- HCP6267I Monitor writer connection to *iucv* ended
- HCP6268E Attempt to connect to Monitor Writer by *application* has been severed
- HCP6269E Macro *name* failed with {return|condition} code *code*
- HCP6270E *iucv* severed the IUCV connection, reason code *code*
- HCP6271A Warning: output file does not contain initial sample configuration records
- HCP6272I Now recording in file *fileid*
- HCP6272I Closing file *fileid*
- HCP6273A Warning: Sample configuration size is too small

## MONWSTOP

---

► MONWSTOP ◄

### Purpose

Use the MONWSTOP command to cause the MONWRITE program to stop processing, close the output file, and exit normally.

### Usage Notes

1. To use the MONWSTOP command, the IUCV server (for example, \*MONITOR) must be specified on an IUCV statement in your directory entry.
2. MONWSTOP is a CMS immediate command.
3. You can also stop MONWRITE from processing by issuing the following CP command to cause an external interrupt:

```
#cp external
```

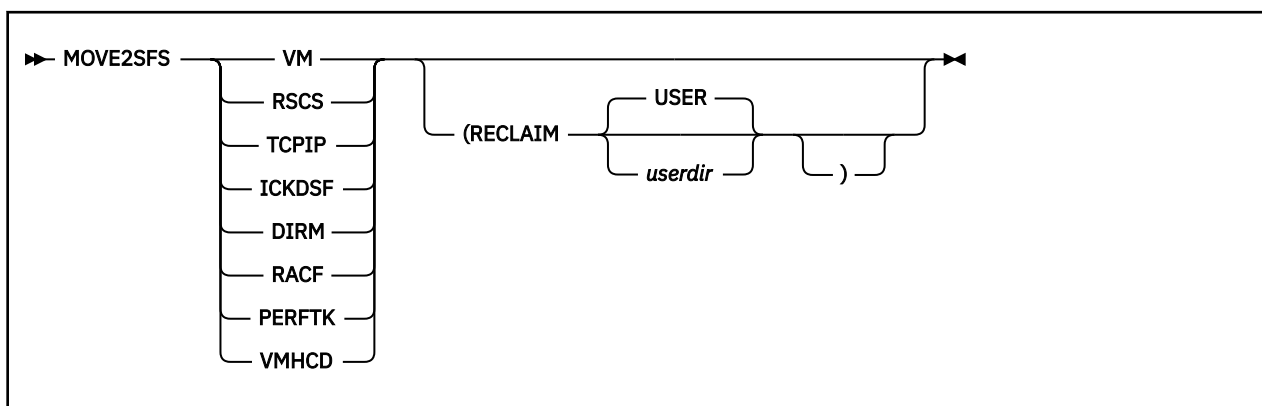
4. If the IUCV server specified on the MONWRITE command is the CP monitor system service, \*MONITOR, and no other users are connected, data collection goes to a pending state.
5. For normal system termination, issue the MONWSTOP command before you shut down the system. You should receive message HCP6265A indicating that MONWRITE has successfully connected to IUCV. If not, use the #CP EXTERNAL command to stop MONWRITE, because MONWRITE may not be ready to recognize the MONWSTOP command. If the MONWSTOP command is processed when MONWRITE is not ready to recognize it, it is ignored by MONWRITE. It is stacked by CMS for future processing—that is, it remains in the program stack until MONWRITE is successfully stopped by other means. If MONWRITE is not ready to recognize the #CP EXTERNAL command as well, CP passes the external interrupt to CMS for processing. See [“EXTERNAL” on page 469](#) for more information.
6. You may use the CMS HX immediate command to stop MONWRITE, but the output file is left in an unpredictable state.

### Messages

- HCP6265A A Monitor writer connected to *iucvserver action*
- HCP6267I Monitor writer connection to *iucv* ended
- HCP6271A Warning: output file does not contain initial sample configuration records



## MOVE2SFS



### Purpose

Use the MOVE2SFS utility to move data from minidisks to the Shared File System servers (SFS) and optionally reclaim the minidisk space. MOVE2SFS creates the subdirectories on the VMSYS file pool that each component needs and then copies the data from the minidisks to the correct subdirectories. The System-Level Software Inventory tables VM SYSRECS and VM SYSAPPS are updated.

### Operands

#### VM

If VM is chosen, all nine components of VM will be copied from minidisks to SFS.

#### RSCS

If RSCS is chosen, the data for RSCS will be copied from minidisks to SFS.

#### TCPIP

If TCPIP is chosen, the data for TCPIP will be copied from minidisks to SFS.

#### ICKDSF

If ICKDSF is chosen, the data for ICKDSF will be copied from minidisks to SFS.

#### DIRM

If DIRM is chosen, the data for DirMaint will be copied from minidisks to SFS.

#### RACF

If RACF is chosen, then the data for RACF will be copied from minidisks to SFS.

#### PERF TK

If PERF TK is chosen, the data for Performance Toolkit for z/VM will be copied.

#### VMHCD

If VMHCD is chosen, the data for VMHCD will be copied from minidisks to SFS.

### Options

#### RECLAIM

reclaims minidisks of moved items by commenting out their entries in the directory specified, bringing the directory online, and detaching the minidisks.

#### *userdir*

is the file name of the directory file. USER is the DEFAULT.

### Usage Notes

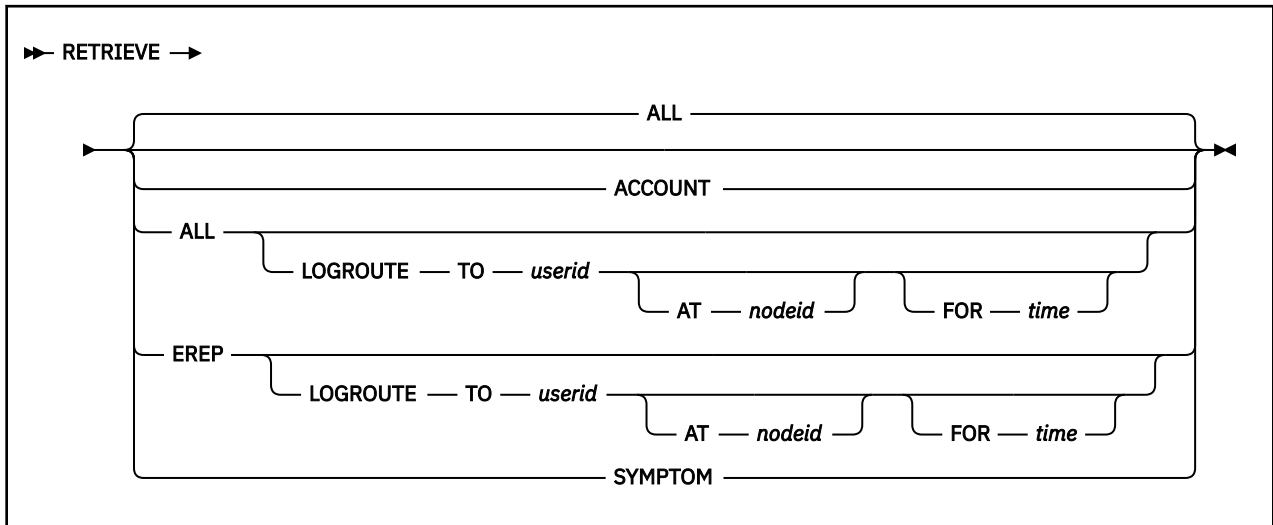
1. The MAINT 193 minidisk must be accessed.

2. You must access the 2CC and the 4CC minidisk in R/W mode.
3. The Software Inventory minidisk (default is the MAINTvrm 51D) must be accessed as the file mode defined in VMFINS DEFAULT and it must be accessed in R/W mode. By default, the Software Inventory minidisk is the MAINTvrm 51D disk and is accessed as file mode D.
4. The VMSYS and VMPSFS file pools must be active.
5. MOVE2SFS will only work if the system is not a member of an SSI cluster or has only one member in the SSI cluster. If there are two or more members in the SSI cluster, the command will not move any products or components.

## Messages

- IUG8300E FILE *filename filetype* NOT FOUND
- IUG8312I ERROR DISPLAYING HELPFILE *name*
- IUG8325E MOVE2SFS cannot be run on an SSI cluster with more than one member
- IUG8342E THE COMMAND *command* FAILED WITH RC=*rc*
- IUG8352E INVALID OPTION(S) / OPERAND(S) *op* SPECIFIED ON THE MOVE2SFS COMMAND
- IUG8353W UNDEFINED PFKEY
- IUG8354W ENTER KEY NOT SUPPORTED
- IUG8376E MOVE2SFS EXEC ENDED IN ERROR
- IUG8392I MOVE2SFS EXEC ENDED SUCCESSFULLY
- IUG8399E COPYFILE FROM THE MINIDISK *mdisk* TO *subdirectory\_name* FAILED WITH RC=*rc*
- IUG8422E ATTEMPT TO QUERY DISK FAILED WITH RC=*rc*
- IUG8431E THE *addr* DISK MUST BE ACCESSED AS *filemode* IN R/W MODE TO RUN *command*
- IUG8437E TOO MANY ARGUMENTS: *options*
- IUG8445E THE {FILEPOOL|DIRECTORY} *filepool\_name* IS NOT AVAILABLE
- IUG8446I THE FOLLOWING COMPONENT WAS ALREADY MOVED TO SFS: *component*
- IUG8449E THE SUBDIRECTORY *subdirectory\_name* COULD NOT BE CREATED
- IUG8450E ACCESS OF {*mdisk|subdirectory\_name*} AT FILEMODE *filemode* FAILED WITH RC=*rc*
- IUG8451W VMFERASE FAILED ON SUBDIRECTORY *subdirectory\_name* WITH RC=*rc*
- IUG8452W VM SYSRECS TABLE WAS NOT UPDATED FOR THE FOLLOWING COMPONENT: *component*
- IUG8453I MOVE OF *component* COMPONENT TO SFS COMPLETED SUCCESSFULLY
- IUG8454E THERE ARE NOT ENOUGH FREE FILEMODES AVAILABLE. TWO ARE REQUIRED
- IUG8455W MOVE2SFS EXEC COMPLETED WITH WARNINGS
- IUG8456I PROCESSING COMPONENT *component*
- IUG8457W VM SYSRECS TABLE WAS ALREADY UPDATED FOR *component*
- IUG8458W *component* IS NOT IN THE VM SYSRECS TABLE
- IUG8465I THE FOLLOWING DISKS FOR COMPONENT *component* HAVE BEEN RECLAIMED: *disk*
- IUG8466I *fn* DIRECT HAS BEEN UPDATED TO COMMENT OUT RECLAIMED MINIDISKS FOR THE MAINT USER ID
- IUG8467I BOTH AVS AND TSAF MUST BE MOVED TO SFS BEFORE THE DISK SPACE CAN BE RECLAIMED
- IUG8470W DETACH OF MINIDISK *mdisk* FAILED WITH RC=*rc*

# RETRIEVE



## Purpose

Use the RETRIEVE utility to collect accounting, I/O error (EREP), and symptom records and to place them in the proper file for later processing and analysis.

## Operands

**ACCOUNT**

**EREP**

**SYMPTOM**

**ALL**

indicates the type of records that you want to collect. Specify ACCOUNT if you want to collect only accounting records; specify EREP if you want to collect only EREP records; and specify SYMPTOM if you want to collect only SYMPTOM records. Specify ALL if you want to collect all of the record types. This is the default.

### LOGROUTE TO *userid*

indicates the user ID to which error records will be routed. LOGROUTE is valid only when EREP or ALL is also specified. If EREP or ALL is specified and LOGROUTE is omitted, the error records are accumulated in CMS files on file mode A. They are not routed to another user ID.

### AT *nodeid*

indicates the node ID to which error records will be routed. If AT is omitted, RETRIEVE processing assumes the LOGROUTE user ID is on the same system.

### FOR *time*

indicates the time interval at which error records will be collected and sent to the specified user. For *time*, specify a whole number from 1 to 1440 indicating the time interval, in minutes. If this parameter omitted, the default is 30 minutes.

## Usage Notes

1. The user issuing the RETRIEVE utility must have a directory entry containing an IUCV directory statement authorizing the virtual machine to connect to the CP system service which supports the type of record being collected. \*ACCOUNT must be specified for ACCOUNT records, \*LOGREC for EREP records, and \*SYMPTOM for SYMPTOM records. Successful completion of this command will cause this user ID to be known to CP (for recording purposes) until a cold start of the system is performed.

## RETRIEVE

2. You can interrupt RETRIEVE processing by entering the EXTERNAL command. You can then decide to stop processing altogether or request a summary of the records collected so far and resume processing by responding to a prompting message issued by the RETRIEVE utility.
3. As records are collected, they are placed in one or more CMS files based upon their type.

**System accounting records** are placed in a file named ACCOUNT \$Ammddy where *mddy* indicates the month, day, and year on which the files were collected.

**Error recording records** are added either to the file XAEREPIO RECORD for outboard type records, or the file XAEREPMC RECORD for inboard type records. The CPEREPA utility needs these files for its processing. Error recording records are also placed in XAEREPSD RECORD when LOGROUTE is specified. For an example of a procedure using the CPEREPA utility to process the collected EREP records, see [Setting Up Virtual Machines to Collect EREP Records in z/VM: CP Planning and Administration](#).

**Note:** Short object recording (OBR) records are not useful for configuration and error management and are not routed by the LOGROUTE parameter.

**Each symptom record** is placed in a separate CMS file whose name is based on the time the symptom record was processed by CP. All symptom record files have the characters SYM as the last three characters of the file type. You can use the CMS VIEWSYM command to select symptom records for viewing based on user-specified criteria. If a symptom record is received that contains unformatted dump-related data supplied by a virtual machine, the portion of the symptom record containing the dump data is not formatted by the VIEWSYM command; it is presented as raw hexadecimal data. The maximum size symptom record supported by z/VM is 4K bytes.

4. At least eight 800-byte CMS disk records must be available on your A-disk and your virtual machine storage size must be at least 512K bytes in size in order to use this command.
5. When LOGROUTE is specified, the user ID and node ID can be either a VM or TSO system.
6. When LOGROUTE is specified, RETRIEVE utility processing accumulates certain error records in a CMS file named XAEREPSD RECORD A. The records collected are ones which should be handled as soon as possible. The records include: long outboard type records for permanent errors on 3380 models D, E, J, and K; long outboard type records for serial switches (9033 and 9032); temporary outboard type records with a fault symptom code of 9F9F; service alert records (type A3); and serial incident records (type A2).

The file is routed to the specified user ID every time the indicated time interval elapses. It is also routed when a service alert or serial incident record is received, even if the time interval has not yet elapsed. After the file is sent, it is erased.

If you used a CP EXTERNAL command to halt RETRIEVE processing, any accumulated records will be routed to the user ID.

7. The RETRIEVE utility does not support recording retrieval to SFS directories.
8. The RETRIEVE utility attempts to periodically close the files which it creates and updates. The committing of the close can be delayed if there are other files on the disk which are open. This can occur when RETRIEVE has been invoked by an EXEC and files were opened but not closed prior to invoking RETRIEVE. It is recommended that all files be closed on the target disk prior to invoking RETRIEVE so that the closes performed by RETRIEVE are not delayed.

## Responses

Response 1:

```
RECORDING RETRIEVAL DISK IS FULL
```

is issued when disk space is no longer available to add records to the various files.

Response 2:

```
RECORDING RETRIEVAL DISK nn% FULL
```

is sent when your disk passes either the 75% or 90% full status. The message is sent only once per threshold.

Response 3:

```
ENTER END OR SUMMARY
```

is the prompting message that you receive in response to an external interrupt.

Response 4:

```
nnnn EREP RECORDS PROCESSED
```

is the response displayed if you are retrieving error records and request a summary. The variable *nnnn* indicates the number of error records that were processed.

Response 5:

```
nnnn ACCOUNTING RECORDS PROCESSED
```

is the response displayed if you are retrieving accounting records and request a summary. The variable *nnnn* indicates the number of accounting records that were processed.

Response 6:

```
nnnn SYMPTOM RECORDS PROCESSED
```

is the response displayed if you are retrieving symptom records and request a summary. The variable *nnnn* indicates the number of symptom records that were processed.

**Note:** If you are retrieving more than one type of record when you request the SUMMARY, one message is issued for each type of record you are retrieving.

Response 7:

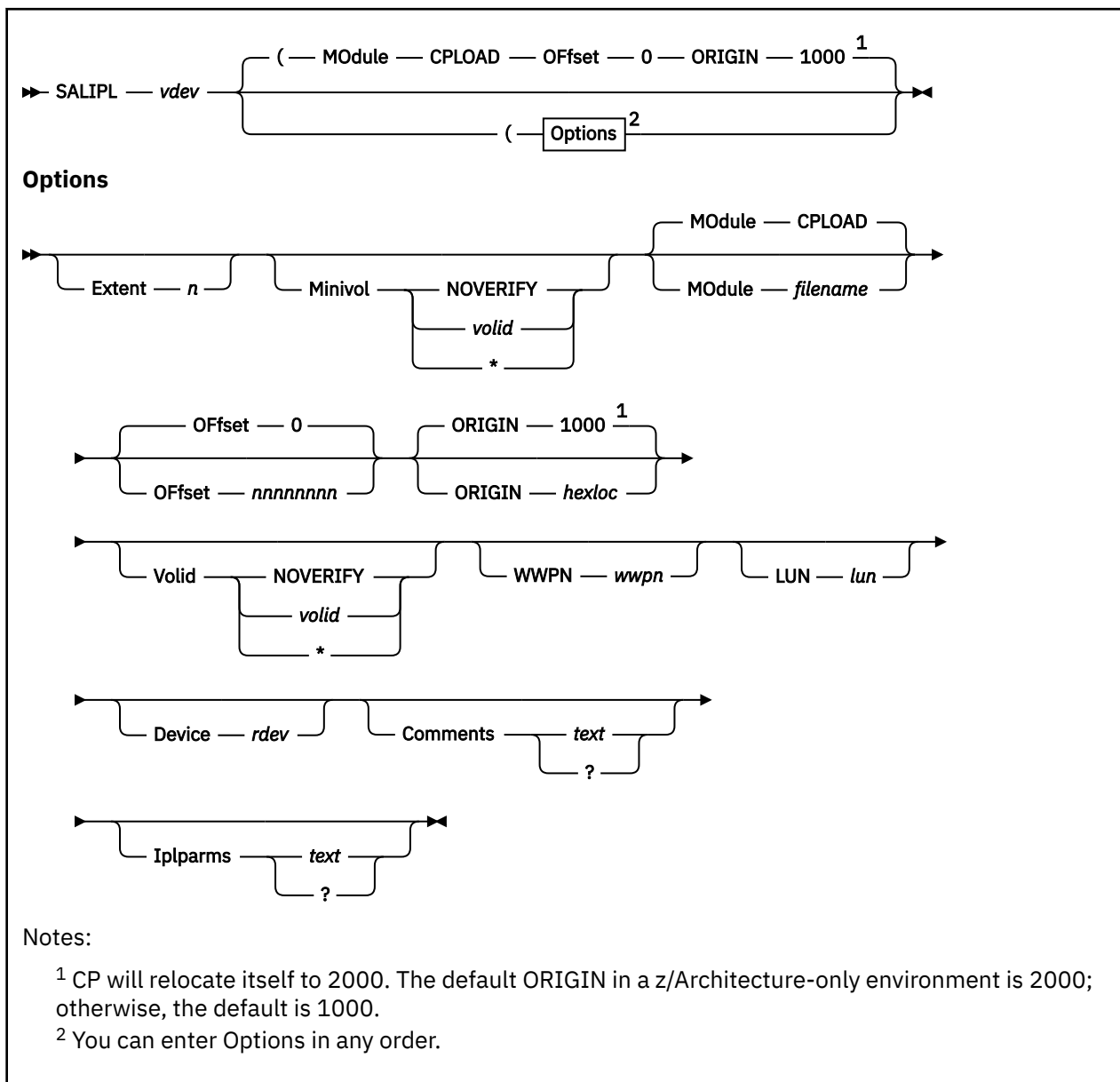
```
RECORDING RETRIEVAL TERMINATED
```

is sent when RETRIEVE ends processing, either upon your request, or because of an error condition.

## Messages

- HCP863E Invalid parameter - *parameter*
- HCP2218E DISK *mode* IS READ-ONLY
- HCP2219E DISK *mode* NOT ACCESSED
- HCP8055E RECORDING RETRIEVAL cannot be performed to filemode *mode* because it is a directory.
- HCP8069E RECORDING RETRIEVAL, RECORD *recording-name nnnn* TOO LARGE[, USER *userid*]
- HCP8070E RECORDING RETRIEVAL *recording-name* WRITE ERROR *rc*[, USER *userid*]
- HCP8081E Attempt to connect to {\*LOGREC|\*ACCOUNT|\*SYMPTOM} system service has failed. Return code *return code*
- HCP8085E An IUCV IPRCODE of *iprcode* was encountered during an IUCV {CONNECT|DCLBFR|RECEIVE | REJECT | REPLY |RTRVBFR|SEVER} function.
- HCP8088I EREP LOGROUTE RECORDS ROUTED TO *userid* AT *nodeid*
- HCP8089I LOGROUTE DEACTIVATED TO *userid* AT *nodeid* FOR *time*
- HCP8090I LOGROUTE ACTIVATED TO *userid* AT *nodeid* FOR *time*

## SALIPL



## Purpose

Use the SALIPL utility to install a copy of the Stand-Alone Program Loader (SAPL) in cylinder zero or, in the case of FBA DASD, in blocks 5 to 207 of a disk.

The SALIPL utility can run under CMS or it can be loaded by SAPL to run stand-alone. There is also a version of SALIPL that can be IPLed stand-alone from tape or card readers. For information on [Running the Stand-Alone Program Loader Creation Utility \(SALIPL\)](#) refer to *z/VM: System Operation*.

When SALIPL writes SAPL to the IPL volume, defaults for the following attributes that SAPL uses are established:

- The location of the CMS minidisk where modules can be found
- The name of the module that is to be loaded by SAPL
- The location in storage at which the module will be loaded, if it is relocatable

- The IPL parameters that are passed to the program being loaded
- Operator instructions that are displayed on the SAPL screen.

## Operands

### *vdev*

specifies the virtual device number (*vdev*) representing the dedicated volume, full-pack minidisk, or regular minidisk on which SAPL is to be installed. The virtual device may be located on an emulated device.

If SAPL is installed on a regular minidisk, it cannot be IPLed in stand-alone mode. This type of SAPL installation is typically used for initializing second level systems and utilities that need to run stand-alone in a virtual machine.

The *vdev* operand may specify an FCP device. If this is done, the WWPN and LUN options must be specified.

## Options

### Extent *n*

specifies which minidisk extent on a CP-formatted volume contains the module to be loaded. *n* is a single decimal number from 1 to 9. 1 refers to the first extent formatted as 'PARM', while 9 refers to the ninth such extent.

**Note:** If you specify EXTENT, you cannot specify OFFSET.

### Minivol

specifies whether SALIPL should check the label of the CMS formatted minidisk from which SAPL will load the module. If MINIVOL is not specified, SALIPL skips the check but displays the minidisk label when installing SAPL.

### NOVERIFY

specifies that SALIPL is not to check for a matching minidisk label and the label of the minidisk should not be displayed in response to the SALIPL invocation.

### *valid*

is a 1- to 6- character minidisk label. If the minidisk identified by OFFSET does not have a matching label, SALIPL issues an error message and no new loader is installed on the target *vdev*. If the minidisk does have a matching label or if *valid* is specified as an asterisk (\*), no minidisk label is displayed in response to the SALIPL invocation.

### Module

specifies the file name of the module that SAPL is to load.

### CPLOAD

is the default module name.

### *filename*

is the file name of the module that SAPL is to load. The file type of the module must be MODULE.

### Offset *nnnnnnnn*

specifies the offset from the beginning of the disk (identified by *vdev*) at which the CMS-formatted minidisk containing the module resides. You can specify up to 10 digits for OFFSET. For CKD and ECKD DASD, the offset is specified as the number of cylinders from the beginning of the disk. For FBA DASD, the offset is specified as the number of blocks from the beginning of the disk.

If you do not specify OFFSET or EXTENT, the default is OFFSET 0.

**Note:** If you specify OFFSET, you cannot specify EXTENT.

### ORIGIN

specifies the address at which SAPL should load the module.

This option is ignored by SAPL if the module that is being loaded has not been generated with the RLDSAVE option. In this case, the module is loaded into storage at the location at which the module was generated.

The entire module, including its required CP data structures, must fit below the 2G line.

When loading CP, the ORIGIN option is accepted, but CP will relocate itself to location X'2000'.

**X'1000'**

is the default address except when SAPL runs in a z/Architecture-only environment, and then the default is 2000.

**hexloc**

is the address at which to load the module.

**WWPN *wwpn***

specifies the world wide port name for a specific path to a SCSI device. The *wwpn* must be a 16-digit hexadecimal number between X'00000000 00000000' and X'FFFFFFFF FFFFFFFF'. WWPN is only valid if the *vdev* operand refers to an FCP device.

**LUN *lun***

specifies the logical unit number for a specific path to a SCSI device. The *lun* must be a 16-digit hexadecimal number between X'00000000 00000000' and X'FFFFFFFF FFFFFFFF'. LUN is only valid if the *vdev* operand refers to an FCP device. Note that the number of digits recognized by the SCSI device may vary by manufacturer, type and model. You must be careful to specify all 16 digits, using *trailing* zeros, as explained in the note below.



**Attention:**

1. *lun* and *wwpn* must be specified as 16 hexadecimal digits. This can be accomplished by specifying two 8-digit tokens, or by specifying one 16-digit token. CMS command invocation may truncate any token to 8 characters, with no warning. For safest operation, specify these arguments as two 8-digit tokens. However, if you know that the CMS command invocation technique uses extended plist (which is done for commands entered at the console or in a REXX EXEC), you can safely specify the value as a single 16-digit token.
2. Because SALIPL cannot know how many digits your device recognizes, you must enter *all 16 digits*. If your SCSI device is identified by fewer than 16 digits, you should use *trailing* zeros to fill out the 16-digit *lun*. Note that if you enter non-zero digits following the digits identifying the SCSI device, those extra digits may be ignored by your device. (For example, if your device recognizes a four-digit *lun*, 5A51888811221122 and 5A51999933443344 would both represent the same device.) This could lead to unintended device sharing and hence might introduce data-integrity exposures.

**Note:** Always identify your SCSI device by its actual *lun* unit number as defined in your storage-area network, padded on the right with zeros.

**Valid**

specifies whether SALIPL should check the volume serial identifier of the CP or CMS-formatted volume on which SAPL is being installed. If VALID is not specified, SALIPL skips the check but displays the volume serial identifier when installing SAPL.

**NOVERIFY**

specifies that SALIPL is to perform no check for a matching volume serial identifier and the volume serial identifier of the *vdev* should not be displayed in response to the SALIPL invocation.

***valid***

is a 1- to 6-character volume serial identifier. If the volume identified by *vdev* does not have a matching *valid*, SALIPL issues an error message and no new loader is installed on the *vdev*. If the volume does have a matching *valid* or if *valid* is specified as an asterisk (\*), no volume serial identifier is displayed in response to the SALIPL invocation.

**Device *rdev***

specifies the address of the device where the minidisk containing the module to be loaded resides. This can be a 5-digit device address, in which case the leading digit specifies the subchannel set



number where the device is located. This subchannel set will be used by z/VM to determine which devices belong in the active configuration. If the multiple subchannel set facility is not installed, the leading digit is truncated and subchannel set zero is used. If this operand is not specified, the minidisk is assumed to be located on the IPL device.

### Comments

specifies up to 320 characters of text information that is to be displayed to the operator if SAPL is given a console address in the load parm when it is IPLed.

#### **text**

is the text to be displayed to the operator.

#### **?**

causes SALIPL to prompt the operator for exactly 4 lines of data to be read from the console. Each line is either truncated to 80 characters or padded with blanks out to 80 characters.

**Note:** When you enter *text* on the command line, COMMENTS must be the last option you specify.

### Iplparms

specifies information that is to be passed to the loaded program by SAPL.

#### **text**

specifies up to 240 characters of text information that is to be passed on to the loaded program by SAPL.


**Note:** When you enter *text* on the command line, IPLPARMS must be the last option you specify.

The format of the IPL parameters depends entirely on the options supported by the loaded program. For a list of IPL parameters supported by CP, see [Passing IPL Parameters in z/VM: System Operation](#).

#### **?**

causes SALIPL to prompt the operator for exactly 3 lines of data to be read from the console. Each line is either truncated to 80 characters or padded with blanks out to 80 characters.

### Usage Notes

1. No special privilege classes are required to run the SALIPL utility. The command issuer must, however, have write access to the volume on which SAPL is to be installed.
2. If you are running SALIPL from an exec, it is suggested that you provide SALIPL with both the label of the volume on which SAPL is to be installed and the label of the CMS-formatted minidisk from which the module is to be loaded. This information is used by SALIPL to ensure that SAL is being installed on the correct volume and is pointing to the correct minidisk on that volume.
3. If both IPLPARMS and COMMENTS are specified with a single '?' as parameters, the user is prompted to enter the text for the options in the same order as they were specified on the command line.
4. If you are IPLing CP from an FCP device (SCSI disk), PDVOL=*raddr* must be specified as an IPL parameter. Any device address that is not in the I/O configuration of the system to be IPLed can be specified for *raddr*. CP requires this parameter in order to build the EDEVICE control block for the SYSRES / PARM disk device during CP initialization.
5.  **Attention:** The number of blocks on FBA DASD (SCSI or not) that SALIPL uses for SAPL has increased. SALIPL writes to blocks 5-207 on CP-formatted DASD. In previous releases, SALIPL wrote to blocks 5-31. You must ensure that no other functions, such as CP directory, warmstart, checkpoint, paging, spooling, or minidisks, are using the area to which SALIPL writes.

On a CMS minidisk with a RECOMP area, SALIPL writes SAPL to blocks 5-207 of the RECOMP area. In previous releases, blocks 5-31 of the RECOMP area were used. In order to accommodate the larger size of SAPL, you may need to increase the size of the RECOMP area on the CMS minidisk.

The following data is in the blocks that SALIPL writes out:

Blocks 5-196

SAPL program

Blocks 197-199	Reserved for IBM Use
Blocks 200-207	SCSI boot records

## Responses

If you want to install SAPL on device 0400, a full-pack minidisk to which you have R/W access, and you want SAPL to use the CMS-formatted minidisk that is 210 cylinders into the full-pack minidisk as the disk from which it will read a module, you enter this command, and see the responses:

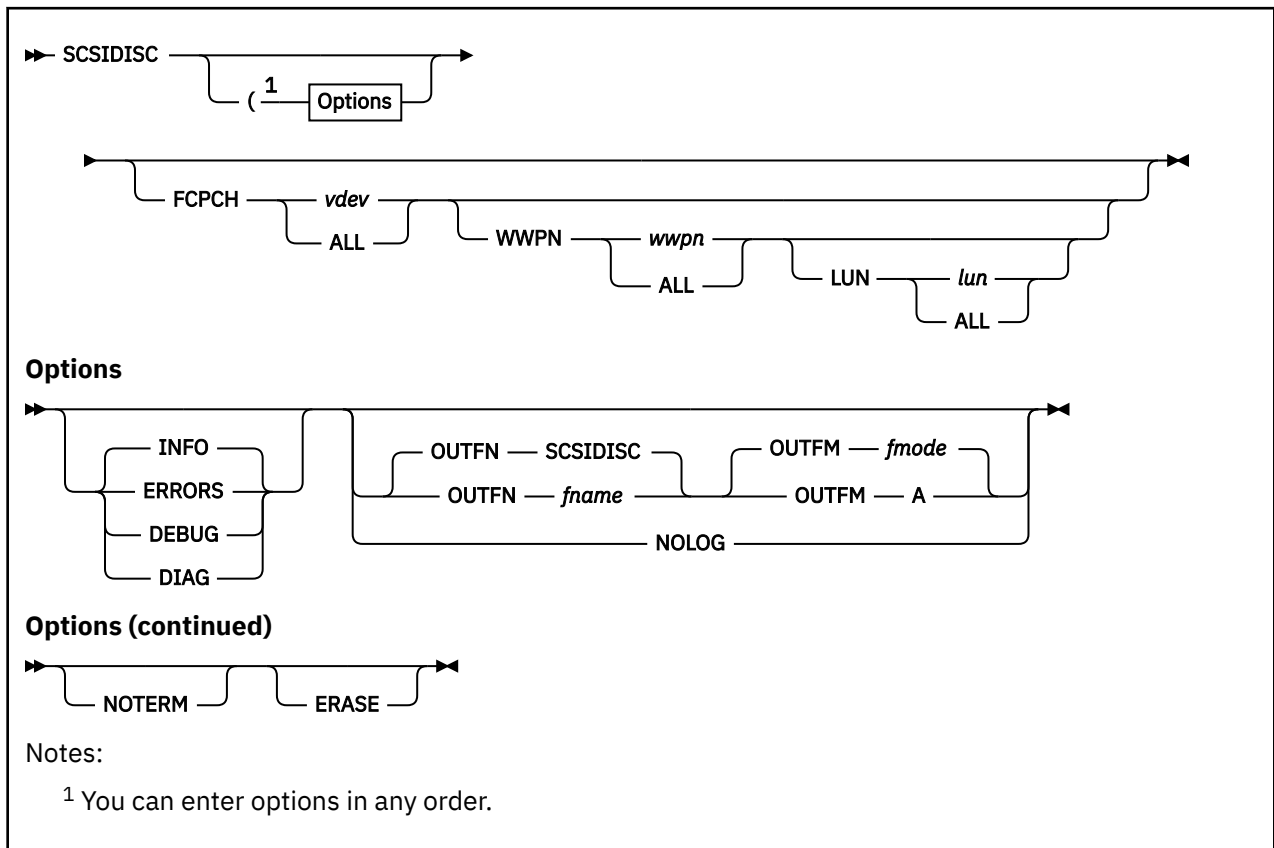
```
salipl 0400 (offset 210 module cpnew iplparms fn=sysnew
VOLUME ID IS ESARES
MINIDISK VOLID AT OFFSET 210 IS PARMD1
Ready;
```

## Messages

- HCP003E INVALID OPTION - {*option*|COMMAND CONTAINS EXTRA OPTION(S) STARTING WITH *option*}
- HCP004E INVALID HEXLOC - *hexloc*
- HCP006E INVALID DEVICE TYPE - *vdev*
- HCP013E CONFLICTING OPTION - *option*
- HCP021E A REAL DEVICE NUMBER WAS NOT SPECIFIED OR IT WAS INVALID.
- HCP022E A VIRTUAL DEVICE NUMBER WAS NOT SPECIFIED OR IT WAS INVALID.
- HCP023E VOLID MISSING OR INVALID
- HCP040E DEVICE *vdev* DOES NOT EXIST.
- HCP352E FATAL DASD I/O ERROR
- HCP393E I/O ERROR ON DISK *vdev*; INCORRECT RECORD LENGTH
- HCP394E I/O ERROR; DISK *rdev* IS BUSY OR HAS INTERRUPT PENDING
- HCP395E PERMANENT I/O ERROR ON DISK *rdev*; SENSE=xxxxxxx
- HCP396E I/O ERROR. THIN STACK FUNCTION=*function* RC=*rc*
- HCP422E THE SAME OPTION WAS SPECIFIED TWICE
- HCP1001E AN OPERAND IS MISSING FOR *option*
- HCP2218E DISK *vdev* IS READ-ONLY
- HCP6207E I/O ERROR; DISK *rdev* HAS UNSOLICITED STATUS PENDING
- HCP6741E DISK *rdev* IS NOT CP OR CMS FORMATTED
- HCP6742E MINIDISK VOLID AT OFFSET *nnnnn* IS *valid1* NOT *valid2*
- HCP6743E MINIDISK AT OFFSET *nnnnn* IS NOT CMS FORMATTED
- HCP6744E VOLUME ID IS *valid1* NOT *valid2*
- HCP6745E BLOCK SIZE IS LESS THAN 4096
- HCP6746E THE PARAMETER FOR THE {IPLPARMS|COMMENTS} OPTION IS TOO LONG.
- HCP6747E WWPN and LUN must be specified for an FCP subchannel
- HCP6794E NO RECOMP AREA DEFINED ON CMS FORMATTED DISK
- HCP6795E RECOMP AREA IS TOO SMALL ON CMS FORMATTED DISK
- HCP6796E IPL PROGRAM AREA NOT ALLOCATED PERM
- HCP6797I MINIDISK VOLID AT OFFSET *nnnnn* IS *valid*
- HCP6798I VOLUME ID IS *valid*
- HCP6799E INVALID EXTENT *nnnnn*
- HCP6800E INVALID OFFSET *nnnnn*

- HCP6801E EXTENT IS NOT VALID WITH A CMS FORMATTED DISK
- HCP6802E EXTENT NOT FOUND IN ALLOCATION MAP
- HCP6803I ENTER UP TO *n* LINES OF {IPLPARMS|COMMENTS}

## SCSIDISC



### Purpose

Use SCSIDISC to return information regarding LUNs that are accessible through virtual FCP devices.

### Operands

#### FCPCH

identifies the virtual FCP device to use. This requires 8 hexadecimal digits to be included to identify the FCP device (generally, 4 zeros and a virtual device number).

#### WWPN

identifies the worldwide port number of a DASD subsystem to use. This requires 16 hexadecimal digits to be included as input. WWPN is valid only when FCPCH is valid.

#### LUN

identifies the logical unit within a DASD subsystem to use. This requires 16 hexadecimal digits to be included as input. LUN is valid only when FCPCH and WWPN are valid.

### Options

#### INFO

displays general information during the program's execution. This is the default level of verbosity.

#### ERRORS

displays the same level of information as the INFO option, with the addition of information displayed by the DEBUG option when a WWPN/LUN produces an error.

**DEBUG**

displays general and diagnostic information during the program's execution. Results of this option are not documented and are subject to change.

**DIAG**

displays diagnostic output that includes the level of information displayed by the DEBUG option, as well as a trace of SCSI function calls. Results of this option are not documented and are subject to change.

**OUTFN**

is the file name of the Outfile and the Logfile. The default is SCSIDISC. File types are fixed as OUT and LOG, respectively. This option is mutually exclusive with NOLOG.

**OUTFM**

is the one-character alphabetic file mode of both the Outfile and Logfile. The default is A. This option is mutually exclusive with NOLOG.

**NOLOG**

suppresses disk output. NOLOG is an alternative to the OUTFN and OUTFM file specifications. This option is mutually exclusive with OUTFN, OUTFM, and NOTERM.

**NOTERM**

suppresses terminal log output. NOTERM can be specified with OUTFN and OUTFM. This option is mutually exclusive with NOLOG.

**ERASE**

deletes any existing output and log files if they previously exist. This program will prompt for user intervention if the files exist and this option is not specified.

**Usage Notes**

1. Although the SCSIDISC utility can be executed by a user of any privilege class, there is no guarantee that the virtual machine will have a virtual FCP device available.
2. You can enter options in any order.
3. The DEBUG option implicitly enables the INFO level of information.
4. The INFO, ERRORS, DEBUG, and DIAG options are all mutually exclusive. INFO is the default. More than one of these options can be specified, but only the most verbose takes priority. The order of increasing verbosity is: INFO, ERRORS, DEBUG, DIAG.
5. Given the potential for producing large volumes of console data, using SCSIDISC with CMS pipelines is not recommended. Output from SCSIDISC is captured on disk in the LOG and OUT files, which can be used as input to customize for further analysis. If console output suppression is desired, use the NOTERM option (rather than PIPE COMMAND EXEC SCSIDISC with no follow-on stage).
6. If SCSIDISC is executed from a pipe, interface text will not be visible. To prevent this scenario, it is recommended that the FCPCH, WWPN, and LUN parameters are all provided when executing from a pipe. This will avoid being prompted by SCSIDISC.

**Responses**

The SCSIDISC utility has the following responses:

Response 1:

If the SCSIDISC OUT file already exists and the ERASE option was not specified, the following response will be issued:

```
Outfile SCSIDISC OUT A already exists. Erase? (Y/N):
```

Note: "SCSIDISC OUT A" may be replaced by the user defined Outfile.

Response 2:

If the SCSIDISC LOG file already exists and the ERASE option was not specified, the following response will be issued:

```
Logfile SCSIDISC LOG A already exists. Erase? (Y/N):
```

Note: "SCSIDISC LOG A" may be replaced by the user defined Logfile.

Response 3:

If no virtual FCP device is specified with the FCPCH operand, the following response will be issued:

```
Please choose a number corresponding to an FCP device, "ALL" to select all
FCP devices or "QUIT"
```

Note: This will be followed by a numbered list of virtual FCP devices that are accessible to the virtual machine.

Response 4:

If no worldwide port number is specified with the WWPEN operand, the following response will be issued once for each virtual FCP device specified:

```
Please choose a number corresponding to a WWPEN, "ALL" to select all WWPENs
or "QUIT"
```

Note: This will be followed by a numbered list of worldwide port numbers that are accessible through the provided virtual FCP devices.

Response 5:

If no logical unit number is specified with the LUN operand, the following response will be issued once for each WWPEN specified:

```
Please choose a number corresponding to a LUN, "ALL" to select all LUNs or
"QUIT"
```

Note: This will be followed by a numbered list of logical unit numbers that are accessible through the provided virtual FCP devices and associated WWPENs.

Response 6:

Once SCSIDISC execution has completed, the following response will be issued:

```
Storage area network analysis complete
```

Response 7:

If the INFO option was specified and all virtual FCP devices were selected, the following response will be issued:

```
Chosen virtual FCP devices=ALL DEVICES
```

Response 8:

If the INFO option was specified and a single virtual FCP device was selected, the following response will be issued:

```
Chosen virtual FCP device=vdev
```

Response 9:

If the INFO option was specified, the following response will indicate the number of WWPENs accessible through some virtual FCP device:

```
For virtual FCP device=vdev number_of_wwpens WWPENs were found
```

Note: "*number\_of\_wwpns* WWPNS were found" will be replaced with "a point-to-point connection was found" if a point-to-point connection is being utilized.

Response 10:

If the INFO option was specified and all WWPNS were selected, the following response will be issued:

```
For virtual FCP device=vdev chosen WWPNS=ALL WWPNS
```

Response 11:

If the INFO option was specified and a single WWPNS was selected, the following response will be issued:

```
For virtual FCP device=vdev chosen WWPNS=wwpn
```

Response 12:

If the INFO option was specified, the following response will indicate the number of LUNs accessible through some virtual FCP device, WWPNS pair:

```
For virtual FCP device & WWPNS wwpn number_of_luns LUNs were found
```

Note: "WWPNS *wwpn*" will be replaced with "point-to-point destination WWPNS *wwpn*" if a point-to-point connection is being utilized

Response 13:

If the INFO option was specified and all LUNs were selected, the following response will be issued:

```
For virtual FCP device vdev & WWPNS wwpn chosen LUNs=ALL LUNs
```

Note: "WWPNS *wwpn*" will be replaced with "point-to-point destination WWPNS *wwpn*" if a point-to-point connection is being utilized.

Response 14:

If the INFO option was specified and a single LUN was selected, the following response will be issued:

```
For virtual FCP device vdev & WWPNS wwpn chosen LUN=lun
```

Note: "WWPNS *wwpn*" will be replaced with "point-to-point destination WWPNS *wwpn*" if a point-to-point connection is being utilized.

Response 15:

If the INFO option was specified, the following response will be issued when a virtual FCP device is reset:

```
Virtual FCP device vdev reset
```

Response 16:

If the INFO option was specified, the following response will be the number of virtual FCP devices found:

```
Number of virtual FCP devices found=number_of_virtual_fcp_devices
```

Response 17:

If the INFO option was specified, the following response will be issued when a virtual FCP device is initialized:

```
Virtual FCP device vdev initialized
```

Response 18:

If the INFO option was specified, the following response will be issued when a virtual FCP device is reinitialized:

```
Virtual FCP device vdev reinitialized
```

**Response 19:**

If the INFO option was specified, the following response will be issued when a WWPN is opened:

```
For virtual FCP device vdev, WWPN wwpn opened
```

Note: "WWPN *wwpn*" will be replaced with "point-to-point destination WWPN *wwpn*" if a point-to-point connection is being utilized.

**Response 20:**

If the INFO option was specified, the following response will be issued when the utility LUN for a WWPN is opened:

```
For virtual FCP device vdev, utility LUN on WWPN wwpn opened
```

Note: "WWPN *wwpn*" will be replaced with "point-to-point destination WWPN *wwpn*" if a point-to-point connection is being utilized.

**Response 21:**

If the INFO option was specified, the following response will be issued when the utility LUN for a WWPN is closed:

```
For virtual FCP device vdev, utility LUN on WWPN wwpn closed
```

Note: "WWPN *wwpn*" will be replaced with "point-to-point destination WWPN *wwpn*" if a point-to-point connection is being utilized.

**Response 22:**

If the INFO option was specified, the following response will be issued when a WWPN is closed:

```
For virtual FCP device vdev, WWPN wwpn closed
```

Note: "WWPN *wwpn*" will be replaced with "point-to-point destination WWPN *wwpn*" if a point-to-point connection is being utilized.

**Response 23:**

If SCSIDISC exits with an error before attempting to open the log file or output file, then no new log file or output file will be created and any previously existing log file or output file will remain. In this event, the following response will be issued after the error message:

```
Logfile and Outfile not updated
```

**Output Files**

Execution of the SCSIDISC utility will result in two files. The first line of both files will indicate the level of SCSIDISC used to generate them. The format of this line is as follows:

```
SCSIDISC LEVEL 0002
```

This level number will start at 0002 and increase with every version of SCSIDISC. Level 0001 is considered to be the SCSIDISC sample utility. The level should increase any time a change is made to the layout of the Outfile or Logfile.

**1. The Outfile**

The Outfile will contain a list of every path available to the set of virtual FCP devices provided. Each entry also includes additional information regarding the LUN accessible through that path.



If a failure occurs on some portion of the path, an entry will be added to the Outfile with any information that was obtained about that path. All remaining fields in the entry will contain repeated dashes (-) for the entire width of that field.

Prior to the first entry in the file, there will be a header labeling all information to follow. The format of the header will be displayed on a single line. It is displayed here on three lines due to formatting restraints.

```
_FCP_CH_      _WWPN_      _LUN_ID_      _UUID_
_VENDOR_ _PROD MODL _SERIAL_ _CODE _BLK_SIZE_ _DISKBLKS_ _LUN_SIZE_
(bytes) _
```

Each label corresponds to the data below it as follows:

<b>Heading</b>	<b>Meaning</b>
FCP_CH	Virtual FCP device composing this path
WWPN	Worldwide port number composing this path
LUN_ID	Logical unit number composing this path
UUID	Universally unique identifier of the referenced LUN
VENDOR	Vendor of the product associated with this LUN
PROD	Product composing this LUN
MODL	Model of the referenced product
SERIAL	Serial number of the storage server
CODE	Product revision level (code level) of the storage server
BLK_SIZE	Number of bytes in each block
DISKBLKS	Total number of blocks residing on the LUN
LUN_SIZE_(bytes)	Total number of bytes residing on the LUN

## 2. The Logfile

The Logfile will contain a list of all the informational messages issued during SCSIDISC execution.

### Messages

- HCP973E No virtual FCP devices are available
- HCP973E Output files already exist
- HCP973E Unknown file error
- HCP974E *operand* is not a valid command line operand
- HCP974E *option* is not a valid command line option
- HCP974E FCPCH has to be given if WWPN and LUN are given
- HCP974E LUN *lun* must be a 16-character hexadecimal value or ALL
- HCP974E OUTFM *filemode* must be a 1-character alphabetic value
- HCP974E OUTFN *filename* must be a 1- to 8-character alphanumeric or an allowed special character value
- HCP974E Virtual FCP device *device* must be an 8-character hexadecimal value or ALL
- HCP974E WWPN and FCPCH have to be given if LUN is given
- HCP974E WWPN *wwpn* must be a 16-character hexadecimal value or ALL
- HCP974E 'NOLOG' cannot be specified with NOTERM
- HCP974E 'NOTERM' cannot be specified with NOLOG

- HCP975I Given virtual FCP device *vdev* ignored because it is not available and therefore any given WWPN or LUN is also ignored
- HCP975I Nameserver error, all WWPNs on virtual FCP device *vdev* may not have been obtained
- HCP975I Virtual FCP device *vdev* ignored
- HCP975I Virtual FCP device *vdev* ignored because the adapter was not able to connect to the fibre channel network
- HCP975I Virtual FCP device *vdev* not initialized
- HCP975I Virtual FCP device *vdev* not reinitialized
- HCP976I Given WWPN *wwpn* on virtual FCP device *vdev* ignored because a point-to-point connection is being utilized and therefore any given LUN is also ignored
- HCP976I Given WWPN *wwpn* on virtual FCP device *vdev* ignored because it was not returned by the nameserver and therefore any given LUN is also ignored
- HCP976I Point-to-point destination WWPN *wwpn* on virtual FCP device *vdev* could not be closed
- HCP976I Point-to-point destination WWPN *wwpn* on virtual FCP device *vdev* could not be opened
- HCP976I Point-to-point destination WWPN *wwpn* on virtual FCP device *vdev* ignored
- HCP976I WWPN *wwpn* on virtual FCP device *vdev* could not be closed
- HCP976I WWPN *wwpn* on virtual FCP device *vdev* could not be opened
- HCP976I WWPN *wwpn* on virtual FCP device *vdev* ignored
- HCP977I Given LUN *lun* on point-to-point destination WWPN *wwpn* and virtual FCP device *vdev* ignored because it was not returned by the utility LUN
- HCP977I Given LUN *lun* on WWPN *wwpn* and virtual FCP device *vdev* ignored because it was not returned by the utility LUN
- HCP977I LUN *lun* on point-to-point destination WWPN *wwpn* and virtual FCP device *vdev* could not be closed
- HCP977I LUN *lun* on point-to-point destination WWPN *wwpn* and virtual FCP device *vdev* could not be opened
- HCP977I LUN *lun* on WWPN *wwpn* and virtual FCP device *vdev* could not be closed
- HCP977I LUN *lun* on WWPN *wwpn* and virtual FCP device *vdev* could not be opened
- HCP977I The utility LUN on point-to-point destination WWPN *wwpn* and virtual FCP device *vdev* could not be opened
- HCP977I The utility LUN on point-to-point destination WWPN *wwpn* and virtual FCP device *vdev* returned an error
- HCP977I The utility LUN on WWPN *wwpn* and virtual FCP device *vdev* could not be opened
- HCP977I The utility LUN on WWPN *wwpn* and virtual FCP device *vdev* returned an error
- HCP978I READ SIZE error for LUN *lun* on point-to-point destination WWPN *wwpn* and virtual FCP device *vdev*
- HCP978I READ SIZE error for LUN *lun* on WWPN *wwpn* and virtual FCP device *vdev*
- HCP978I INQUIRY error for LUN *lun* on point-to-point destination WWPN *wwpn* and virtual FCP device *vdev*
- HCP978I INQUIRY error for LUN *lun* on WWPN *wwpn* and virtual FCP device *vdev*
- HCP979I Disk write failure, EXECIO\_RC=*return code*

# SDINST

► SDINST ◄

## Purpose

Use the SDINST utility to build a configuration file used for a stand-alone dump containing device addresses for dump IPL DASD as well as additional dump output devices.

## Usage Notes

1. The user must have a R/W minidisk at device number 191 and accessed as file mode A. The configuration file is written there.
2. The SDINST utility creates a file called SDINST DUMPCONF on the user's A-disk. If the file is found at runtime, the user can choose to continue to install stand-alone dump using the existing file, or continue normal configuration and create a new file.  
  
SDINST also creates a dump status file called SDINST CMSUT1 on the user's A-disk. The file contains information for SDINST's DASD formatting routine. This file is recreated each time SDINST is run and can be erased by the user if desired.
3. ECKD or SCSI DASD can be used as dump devices, however devices must be all ECKD or all SCSI.
4. There is a maximum of 30 DASD, including the IPL DASD.
5. The SDINST utility expects a virtual reader device at virtual device number 00C and a virtual punch device at virtual device number 00D.
6. The SDINST utility requires at least 512 MB of virtual storage in order to load its DASD formatting program.
7. The SDINST utility requires that your virtual machine does not have any existing class N reader files.
8. The SDINST utility requires the following files, all of which have a filetype of IMAGE: SADU $nn$ , SSPI $nn$ , SSPK $nn$ , and SSPP $nn$ . The  $nn$  is a version number that SDINST supplies and with which it is compatible. In a standard z/VM installation, the MAINT 400 disk contains these files and should be accessed before running SDINST. The SDINST utility also looks for these files on the 191 disk before the 400 disk, so that service can be tested before it is installed. Files with matching names on other accessed disks are ignored.
9. The first dump device is formatted to contain the stand-alone dump program and is made IPLable. It is required to be large enough to hold the program plus some minimal space for a dump. For dumping a first-level system (LPAR), each dump device is an entire 3390 or SCSI LUN. Currently, a 3390 used by stand-alone dump is limited 65520 cylinders when allocated on an IBM TotalStorage DASD subsystem. That is approximately 45 GB of data. The maximum SCSI device support is limited to about 1 TB because VM's EDEVICE support is used when loading the dump.  
  
When using the stand-alone dump program to dump a second-level system, minidisks may be used. The minidisk used for the IPL device must be at least 75 cylinders. Minidisks for additional dump devices could be as small as 1 cylinder but that is not a reasonable size for dump devices. The minimum SCSI LUN sizes are equivalent: 53 MB for the IPL device and 0.7 MB for additional dump devices.
10. For additional instructions on creating a stand-alone dump utility on ECKD and SCSI devices, taking a stand-alone dump, and processing a stand-alone dump, see Chapter 12, "The Stand-Alone Dump Utility", in *z/VM: CP Planning and Administration*.

## Examples

Typical use of the SDINST utility with no pre-existing configuration file using one 3390 DASD:

```

>>> SDINST
HCPSDI8682I STAND-ALONE DUMP INSTALLER Vd.r
ENTER 'QUIT' AT ANY PROMPT TO TERMINATE EXECUTION.
HCPSDI8654A PLEASE ENTER THE VIRTUAL DEVICE ADDRESS OF THE IPL DASD
555
HCPSDI8655A PLEASE ENTER THE REAL DEVICE ADDRESS FOR THIS DASD
OR ENTER Y TO USE 0555
y
HCPSDI8674A DASD HAS EXISTING VOLID DSD555.
DO YOU WISH TO CONTINUE? (Y/N)
y
HCPSDI8658A ARE THERE ANY ADDITIONAL DUMP DASD TO SPECIFY? (Y/N)
n
SSPK73 IMAGE F1 F 80 113581 1738 yyyy-mm-dd hh:mm:ss
SSPP73 IMAGE F1 F 80 1 1 yyyy-mm-dd hh:mm:ss
SSPJ73 IMAGE F1 F 80 76478 1494 yyyy-mm-dd hh:mm:ss
SADU73 IMAGE F1 F 80 15655 306 yyyy-mm-dd hh:mm:ss

CP SPOOL PUNCH CL N TO * NOHOLD NOEOF NOKEEP
RDR FILE 1146 SENT FROM MYUSERID PUN WAS 1146 RECS 114K CPY 001 N ...
RDR FILE 1147 SENT FROM MYUSERID PUN WAS 1147 RECS 0001 CPY 001 N ...
RDR FILE 1148 SENT FROM MYUSERID PUN WAS 1148 RECS 076K CPY 001 N ...

CP SPOOL RDR CL N NOHOLD EOF
SPOOL RDR OPTIONS HAVE BEEN CHANGED TO CLASS N, NOHOLD AND EOF.
IF FOLLOWING IPL IS SUCCESSFUL, YOU WILL NEED TO RESTORE SPOOL
RDR OPTIONS MANUALLY BECAUSE CONTROL DOES NOT RETURN TO THIS EXEC.

EVERYTHING IS READY. ANSWER Y TO CONTINUE AND IPL THE DASD FORMATTING
PROGRAM. ANY OTHER RESPONSE EXITS WITHOUT DOING THE IPL.
y
CP IPL 00C CLEAR PARM loglevel=0 sspmod=SADU73:191
HCPSDI8660I SETTING DEVICE 0191 ONLINE.
HCPSDI8662I FORMATTING 3339 CYLINDERS ON DEVICE 0555.
HCPSDI8664I WRITING PARTITION TABLE TO DEVICE 0555.
HCPSDI8666I CREATING FILE SYSTEM FOR IPL PARTITION ON DEVICE 0555.
HCPSDI8668I INSTALLING STAND-ALONE DUMP TO IPL PARTITION ON
DEVICE 0555.
HCPSDI8670I STAND-ALONE DUMP INSTALLATION IS COMPLETE.
YOU MAY RE-IPL CMS.

```

## Messages

- HCP002E Invalid operand - *operand*.
- HCP1004E *operand* is not a DASD volume or is an invalid DASD type.
- HCP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU *nn*.
- HCP6741E Disk *operand* is not CP or CMS formatted.
- HCP8219E VDEV must be a hex value - *operand*
- HCP8225E Device *operand* does not exist
- HCP8227E Device *operand* is not R/W
- HCP8625E Image file mismatch.
- HCP8639E This version of the SDINST utility requires hardware features not available on this processor.
- HCP8640E Device *vdev* is not a *devtype*
- HCP8641E Insufficient virtual storage to run the program that formats the stand-alone dump DASD. You need at least 512 MB to proceed.
- HCP8642E This process can not proceed if you already have Class N RDR files.
- HCP8643E STATE error *rc* for required file *file*.
- HCP8644E PUNCH error *rc* for file *file*.
- HCP8645E Error *rc* for command *command*.
- HCP8647E Device *vdev* must have a label.
- HCP8648E THE SDINST UTILITY DOES NOT RUN IN A VIRTUAL MACHINE THAT IS IN ESA/XC OR Z/XC MODE.
- HCP8649E Error *rc* when obtaining device size.

- HCP8650E SDINST requires a 191 minidisk accessed R/W as file mode A.
- HCP8651E *vdev* is a {3390|SCSI} DASD, but IPL DASD is {3390|SCSI}. All DASD must be the same type.
- HCP8652E DASD at virtual address *vdev* is not large enough to install stand-alone dump.
- HCP8653E Volume ID of *vdev* is *valid* instead of expected *valid*.
- HCP8654A Please enter the virtual device address of the {IPL|additional dump} DASD.
- HCP8655A Please enter the real device address for this DASD or enter Y to use *rdev*.
- HCP8656A Please enter the 16 hexadecimal digit WWPN for this DASD.
- HCP8657A Please enter the 16 hexadecimal digit LUN for this DASD.
- HCP8658A Are there any additional dump DASD to specify? (Y/N)
- HCP8659E Attempted to define more than 30 dump devices.
- HCP8660I Setting device *devno* online.
- HCP8661E Could not bring device *devno* online.
- HCP8662I Formatting xxxxxx cylinders on device *devno*.
- HCP8663E There was a problem formatting device *devno*.
- HCP8664I Writing partition table to device *devno*.
- HCP8665I There was a problem writing partition table to device *devno*.
- HCP8666I Creating file system for IPL partition on device *devno*.
- HCP8667E There was a problem creating the file system for the IPL partition on device *devno*.
- HCP8668I Installing stand-alone dump to the IPL partition on device *devno*.
- HCP8669E There was a problem installing stand-alone dump to the IPL partition on device *devno*.
- HCP8670I Stand-alone dump installation is complete. You may re-IPL CMS.
- HCP8671A Existing File SDINST DUMPCONF A found. Enter O to overwrite, C to continue using the existing file or anything else to quit.
- HCP8672I Stand-alone dump installation aborted.
- HCP8673E Device type for *vdev* *vdev* is not as expected.
- HCP8674A DASD has existing *valid* *valid*. Do you wish to continue? (Y/N)
- HCP8675E SDINST failed because no file modes are available.
- HCP8676E Existing file *file* is not a valid stand-alone dump configuration file.
- HCP8677E Existing file *file* was created with an older version of the SDINST utility and is no longer compatible.
- HCP8678E Existing file *file* was created with a newer version of the *sdinst* utility. Please use the newest version of SDINST.
- HCP8679E There was a problem with the DASD specified starting on line *line*.
- HCP8680E Parsing failed on line *line*. Check previous error message.
- HCP8681E There can be only one IPLDASD statement in a stand-alone dump configuration file.
- HCP8682I Stand-alone Dump Installer *Vversion* enter 'QUIT' at any prompt to terminate execution.
- HCP8683E VDEV must be 1 to 4 digits - *vdev*.
- HCP8684E Size of the SCSI device *devno* can not be larger than 2,147,483,640 512-byte blocks.
- HCP8684E Size of the 3390 device *devno* can not be larger than 65520 cylinders.
- HCP8685E *operand* value missing.
- HCP8686E RDEV value not valid - *operand*. It must be a 1 to 4 digit hex number.
- HCP8687E WWPN must be 16 digits - *operand*
- HCP8688E WWPN must be a hex value - *operand*
- HCP8689E LUN must be 16 digits - *operand*

## SDINST

- HCP8690E LUN must be a hex value - *operand*
- HCP8691E VDEV *vdev* cannot be accessed at file mode A.
- HCP8692E *operand* was already specified.
- HCP8693E IPLDASD statement must be the first configuration statement in file *file*.
- HCP8694E The stand-alone dump DASD formatting module is incompatible with this version of the SDINST utility.
- HCP8695E Could not read configuration data from 191 disk.
- HCP8696E An unexpected error has occurred and stand-alone dump installation has been aborted. You may re-IPL CMS.
- HCP8697E The stand-alone dump DASD formatting module encountered invalid configuration data.
- HCP8698E *keyword* keyword on line *line* is not valid when device type is *devtype*.
- HCP8699E Found second occurrence of *keyword* keyword on line *curr* for statement starting on line *line*.

## SECTABLE



### Purpose

Use the SECTABLE utility to associate a line of installation-defined text with a RACF-defined security label (SECLABEL). When a system printer is started with a particular security label, the associated line of text will be printed on the header and trailer page of each file printed on that printer. Security labels are assigned only when multilevel security (MLS) is active.

### Operands

#### STORE

alters the human-readable text associated with each SECLABEL listed in the file SECTABLE FILE. The first copy of SECTABLE FILE found in the CMS file search order will be used.

To use the STORE operand, you must have READ access to the DIAG0A0.HRTSTORE profile in the VMCMD class. If protection for DIAGNOSE X'A0' has been turned off, then you must have class A or B privilege.

If you omit the STORE operand, no special privileges are required as the file will only be checked for correct syntax.

### Usage Notes

1. By default, the SECTABLE utility is located on the system tools disk (MAINT 193).
2. SECTABLE FILE is a variable-format (RECFM V) file with a logical record length between 10 and 141 bytes. The size of the file cannot exceed 65,536 bytes (64 KB). The format of SECTABLE FILE is:

Column	Content
1 - 8	SECLABEL character string
9	Required blank space
10 - 141	Human-readable label (1 to 132 characters long)

Ensure that only authorized administrators have write access to the minidisk containing SECTABLE FILE.

3. SECTABLE STORE must be run each time the system is IPLed. Alter the PROFILE EXEC for AUTOLOG2 to issue the SECTABLE STORE command *before* any other service machines or system printers are started.
4. A system printer has a default security label of NONE and will not print until SECTABLE STORE has been issued and a valid SECLABEL has been specified on the CP START command. The SECLABEL must be listed in SECTABLE FILE.
5. For more information on RACF authorization for DIAG0A0.HRTSTORE, see [Protecting the DIAGNOSE X'A0' Subcodes](#) in *z/VM: RACF Security Server Security Administrator's Guide*.

### Messages

- HCP8601I All SECTABLE file records were found to be valid.

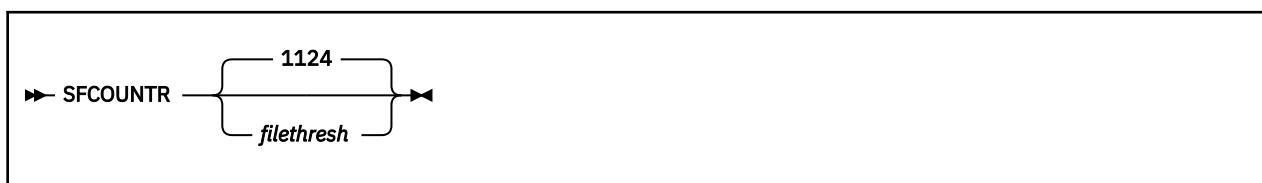
## SECTABLE

- HCP8602I SECLABEL/HRL correlation table has been successfully stored. [RC=0]
- HCP8603E Invalid operand specification on SECTABLE invocation. [RC=2]
- HCP8604E Error from CMS *macro* macro with return code *retcode*. [RC=4]
- HCP8605E SECTABLE FILE must have LRECL not < 10, LRECL not > 141, and RECFM = V. [RC=6]
- HCP8606E SECTABLE FILE is too large - CP virtual buffer will be > 64K bytes. [RC=8]
- HCP8607E Record *n* causes CP virtual buffer to overflow 64K bytes.
- HCP8608E The following represent invalid SECTABLE FILE records:
- HCP8609E A total of *n* records were found to be invalid. [RC=10]
- HCP8610E Condition code of *condcode* upon return from DIAGNOSE X'A0' subcode X'34' processing. [RC=12]
- HCP8611E Severe error occurred during DIAGNOSE X'A0' subcode X'34' processing.

To find the description of a message, see [z/VM: CP Messages and Codes](#).



## SFCOUNTR



### Authorization

Privilege Class: D

### Purpose

Use the SFCOUNTR utility to create a list of all of the users who own more than a specified number of spool files.

This utility is specifically designed for single-configuration virtual machines (USERS), not multiconfiguration virtual machines (IDENTITY users), in a single system image (SSI) cluster. This utility is needed because USERS are limited to fewer than the maximum 9,999 per user files created on and owned by each SSI cluster member. The limit depends on the number of members for which the cluster is enabled. If the installation is increasing the maximum number of members (from 4 to 8, for example), the spool file limit per user per member decreases.

The main use of this utility is during a change of an SSI cluster from having 4 members to having up to 8 members. When this change is made on the SPOOL\_MEMBERS operand of the SSI\_CONTROLS statement in the system configuration file and the system is restarted, users can have a maximum of 1,249 spool files created on and owned by each member. This utility can be used to determine ahead of time if there might be any users with more than the new maximum. For more information, see usage note “2” on page 2241.

### Operands

#### *filethresh*

is the threshold number of spool files owned by a user above which the user will be included in the list. This operand can be specified as any integer from 1 to 9999. The default is 1124, which is 90% of the maximum number of files a user can have in an SSI cluster that is enabled for 8 members.

### Usage Notes

1. This utility ignores logged-on multiconfiguration virtual machines (MCVMs). If an MCVM is not logged onto this member, SFCOUNTR will not recognize it as such and will include it in the report. However, it can be ignored, because MCVMs are always allowed to have the full 9,999 files.
2. A spool file is owned by a cluster member if it was created on that system. The creating system assigns a spool ID and allocates spool space for the file on its CP-owned devices. This utility returns the number of spool files owned by a user. For a user logged onto this member, the number includes files created on other cluster members. Only the files owned by this member are relevant to the question of whether a user will have too many files when the SPOOL\_MEMBERS statement is changed. However, if the reported number is below 1,249 for all users, this member can safely be switched to the SPOOL\_MEMBERS 8 setting.
3. This utility must be run from a privileged user, that is, with certain CP command privileges along with class G. The privileged commands issued and their IBM privilege classes are:

+ QUERY MAXSPOOL SYSTEM	Class D
+ QUERY RDR SYSTEM ALL	Class D
+ QUERY PRT SYSTEM ALL	Class D
+ QUERY PUN SYSTEM ALL	Class D

```

+ QUERY TRF SYSTEM|userid ALL Class A, C, D, or E
+ QUERY FILES SYSTEM|userid Class D
+ LOCATE userid Class C or E
+ DISPLAY Host Class C or E

```

## Responses

The output from SFCOUNTR is a file sent to your A-disk. The file is called *mmddhhmm* SFCOUNTR. The file type is always SFCOUNTR. The file name is an indication of when the utility was run and is of the form *mmddhhmm*, where:

### **mm**

is the month

### **dd**

is which day of the month

### **hh**

is the hour

### **mm**

is the minute

If the utility is run more than once in the same minute, the earlier output files are replaced and only the latest output file from that minute is retained.

The output file will look something like the example below if there are users whose spool file count is over the limit.

The users are listed in descending order by number of files owned.

Note that the second line of the output contains the time and date the utility was run in the user's DATEFORMAT setting. The following example shows the line when the user's DATEFORMAT is set to SHORTDATE.

```

SFCOUNTR data collected by OPERATOR AT VM1
TIME IS 11:32:58 EDT TUESDAY 07/13/22
Threshold was set to 1124 files.
USER FILES
=====
USERABC 2304
USERXYZ 1518
WHOEVER1 1125

```

The output file will look something like the example below if there are no users whose spool file count is over the limit. The last line shows the number owned by the user with the most spool files.

Note that the second line of the output contains the time and date the utility was run in the user's DATEFORMAT setting. The following example shows the line when the user's DATEFORMAT is set to FULLDATE.

```

SFCOUNTR data collected by OPERATOR AT VM1
TIME IS 13:31:11 EDT TUESDAY 07/13/2022
Threshold was set to 1124 files.
No users had more than 518 spool files.

```

## Messages

- HCP002E Invalid operand - *operand*
- HCP263E Too many operands were supplied for this command.
- HCP6651E Required privilege is not available for: *command*.
- HCP8250E SFCOUNTR REQUIRES A R/W A-DISK

# SPORDER

---

► SPORDER ◄

## Purpose

Use the SPORDER utility to put your virtual reader files in chronological order.

## Responses

SPORDER has no normal response; successful completion results in return code 0.

Response 1:

```
No RDR files to order
```

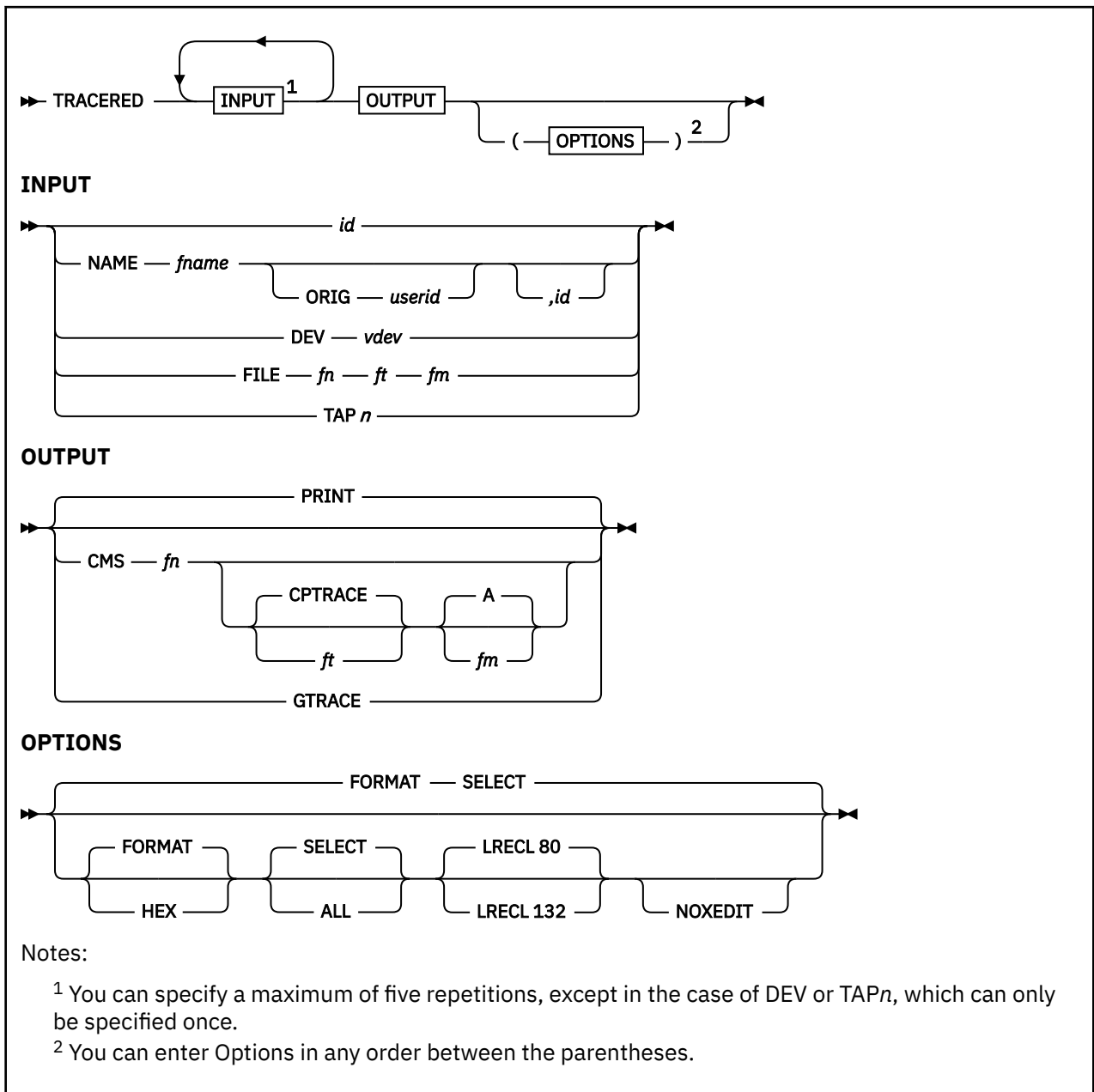
is the response sent if the originator of SPORDER has no files in their virtual reader (the return code is 4).

Response 2:

```
Return code nn from CP Q FILES
```

is the response sent to the originator if an error occurs when SPORDER issues the CP Q FILES command (the return code is 98).

# TRACERED



## Purpose

Use the TRACERED utility to process trace data recorded by the TRSOURCE command or to process CP trace data. CP trace data includes system trace files, data captured in a CMS file by the TRSAVE subcommand, and system trace table records contained on tape or in system trace files. The TRSOURCE command data can be merged with existing CP or TRSOURCE data as follows:

- TRSOURCE data can be merged with CP data, or other TRSOURCE DATA, to generate CP data.
- TRSOURCE data can be merged with other TRSOURCE data to generate a consolidated output file in chronological sequence. Refer to [“TRSOURCE” on page 2018](#) for further information on TRSOURCE, as well as creating trace files.

## Operands

### *id*

specifies the primary input to TRACERED as a system trace file containing CPTRACE or TRSOURCE data. The *id* operand is a spool ID, a decimal number from 1 to 9999.

### NAME

specifies the primary input to TRACERED from one or more files containing trace data collected using SET CPTRACE or TRSOURCE commands. You may identify files by file name, origin ID, and spool ID. Trace file names are assigned by the TRSOURCE and TRSAVE commands. Within a set of traces, you may select trace data for a particular virtual machine by specifying an origin user ID. Further selectivity is allowed by selecting individual trace files by spool ID. The *fn* operand specifies the file name assigned by the TRSAVE or TRSOURCE command. The ORIG *userid* operand identifies, for guest type traces, files containing an individual virtual machine's trace data. This enables an individual member of a VMGROUP trace to be specified on the TRACERED command line. The *id* operand identifies an individual system trace file created with file name *fn*, and it is specified as a spool ID, a decimal number from 1 to 9999.

A total of five system trace files, CMS files, or both can be merged including only one CP TRACE data file. The output is merged by time-of-day and formatted as it is written to a CMS file or the virtual printer as designated by this command.

### DEV *vdev*

specifies the primary input to TRACERED as a tape containing CP trace table data at virtual device address *vdev*.

### FILE *fn ft fm*

specifies a file containing data (for example, TRSOURCE data extracted from a dump with the TRSAVE command) to be processed by TRACERED.

### TAP*n*

specifies the primary input to TRACERED as a tape containing CP trace table data at symbolic tape address TAP*n*. See the FILEDEF command in *z/VM: CMS Commands and Utilities Reference* for a list of supported TAP*n* values.

### PRINT

designates that the output of the TRACERED command goes to your virtual printer. This is the system default.

### CMS

indicates that the output of the TRACERED command should be directed to a CMS file of the file name, file type, and file mode you specify. If you do not specify a file type, the default is CPTRACE. If you do not specify a file mode, the default is A. If you specify this keyword, you will be put into XEDIT when the command completes successfully.

### GTRACE

specifies that the output will be in the form of an OS QSAM file. Only guest virtual machine recorded trace events created by TRSOURCE in either BLOCK mode or EVENT mode are output to this file.

## Options

### FORMAT

specifies that each selected trace entry should appear in a long format. Three lines per trace entry are formatted for CP trace table entries. The number of lines of output for TRSOURCE recorded entries is determined by a formatting routine based on the length of the trace record and the output LRECL value. When the formatting routine is not supplied, the data is in hexadecimal form. The difference between HEX and FORMAT processing is left to the exit routine to define. FORMAT is the default value. Some products may provide only one format, that being either a short (HEX) or long (FORMAT) format for their guest trace data.

The following records are supported for the format option:

- CP trace table records
- CP records of type DATA, IO, and LAN

## TRACERED

- Guest trace records providing a formatting routine that is accessible by TRACERED.

Other guest trace records will be formatted in hex output.

### HEX

stipulates that the hexadecimal (or short) format should be used for the selected entries. CP trace table data output is one line per trace entry. The mixture of hexadecimal and interpreted output is determined by the exit routine for guest trace data. Some products may provide only one format, either a short (HEX) or a long (FORMAT) format for their guest trace data. You may want to use this option when you want output to go to a CMS file and limited DASD space precludes the use of the FORMAT option.

The following records are supported for the format option:

- CP trace table records
- CP records of type DATA, I/O, and LAN
- Guest trace records providing a formatting routine that is accessible by TRACERED.

Other Guest trace records will be formatted in hex output.

### SELECT

signifies that you want selection criteria to be applied to the trace entries. You will be prompted for the selection criteria you want. This is the system default.

### ALL

indicates that all trace entries from the specified input tape or system trace file are to be selected for output. You will not be prompted for selection criteria.

### LRECL 80

### LRECL 132

specifies the record length maximum for the output file. When not specified, CMS files default to 80 bytes and print files default to 132 bytes.

### NOXEDIT

avoids entering an XEDIT session when the CMS keyword is specified on TRACERED. This option may be useful if the CMS output file created is too large to XEDIT.

**Selection Criteria for the SELECT:** If you choose or let the system default to the SELECT option for the TRACERED command, you receive the following prompt:

```
ENTER TRACE ENTRY SELECTION CRITERIA, NULL LINE TO END  
SELECTION, OR QUIT TO END TRACERED COMMAND
```

If you enter a null line, **no** selection criteria are applied. As a result, everything will be traced.

If you enter selection criteria, you will continue to receive this prompt until you enter a null line, thus indicating that entry of selection criteria is complete. Acceptable replies to the prompt are:

1. Any combination of the following keywords and values:

- **CODE xxxx xxxx . . .**

You may enter up to 16 four-digit hexadecimal values representing the trace table codes to be selected during data reduction of the input tape or system trace file. Leading zeros are required. The last two digits of any code may be specified as the character string \*\* or \*. This indicates that you want a range of trace table codes. For instance, 15\*\* or 15\* would select all trace table entries beginning with the hexadecimal digits 15. CODE is valid for CP trace table data only.

- **VMDBK xxxxxxxx xxxxxxxx . . .**

You may enter up to 16 hexadecimal values that represent the VMDBK addresses to be used during data reduction of the input tape or system trace file. Leading zeros are not required. Each address specified must be in the range of X'0' to X'7FFFF000' and on a page (4 KB) boundary. VMDBK is valid only for CP trace table data.

- **RDEV xxxx xxxx . . .**

You may specify up to 16 hexadecimal values that represent the RDEV numbers to be used during data reduction of the input tape or system trace file. Leading zeros are not required. Each device number specified must be in the range of X'0' to X'FFFF'. RDEV is valid only for CP trace table data.

- **VDEV xxxx xxxx . . .**

You may specify up to 16 hexadecimal values that represent the VDEV numbers to be used during data reduction of the input tape or system trace file. Leading zeros are not required. Each device number specified must be in the range of X'0' to X'FFFF'. VDEV is valid only for CP trace table data.

- **CPU xxxx xxxx . . .**

You may specify up to 16 hexadecimal values that represent the processor addresses to be used during data reduction of the input tape or system trace file. Leading zeros are not required. Each processor address specified must be in the range of X'0000' to X'004F'. CPU is valid only for CP trace table data.

- **TIME mm/dd/yy hh:mm:ss mm/dd/yy hh:mm:ss**

You may designate a start/stop date/time for the data reduction of the input trace entries. If this option is selected, only trace entries created within the designated date and time range will be processed. You may select only one date/time range. All the TIME parameters may be input on the same line.

**Note: TIME** is the only valid selection criteria if TRSOURCE entries are being processed.

2. CMS – To enter the CMS subset mode.
3. HELP – To invoke the help panel for the TRACERED command.
4. QUIT – To end the TRACERED command.
5. A null line – To indicate that you are done entering selection criteria and want to end prompting.

If your immediate response to the prompt is a null line, then all trace entries are selected.

## Usage Notes

1. The *filename* of the NAME operand designates a set of system trace files created by one invocation of the CP TRSAVE command. Multiple system trace files may exist with the same file name. If you specify a file name as NAME *fn*, TRACERED processes all system trace files with that file name, beginning with the oldest. If you specify NAME *fn1 ,id1*, TRACERED processes all system trace files with that file name beginning with the spool ID *id1*. If you specify the *id* operand, TRACERED will process that system trace file.
2. Avoid using TRACERED keywords for file names and file types as part of the file id specified with the "FILE" operand.
3. TRACERED uses the first 2 bytes of each TRSOURCE created entry record as the length of that entry.
4. When more than one tape created by the same TRSAVE command is to be mounted and processed, they must be processed in chronological order.
5. Use the QUERY TRFILES command to display information about system trace files you own, including spool ID, file name, and the time of creation.
6. A total of five system trace files, CMS files, or both can be merged. However, only one CP trace table file or tape may be included. Therefore, you may specify one of the following:
  - One CP trace table file with up to four TRSOURCE trace files
  - One tape with CP Trace table and up to four TRSOURCE trace files
  - Up to five TRSOURCE trace files.

You cannot merge a CP tape or tapes with a CP trace table file.

7. The following guidelines apply to your responses to the prompt for selection criteria when **CP trace table data** is processed:
  - Each time the prompt appears, you may specify any or all of the six selection types: CODE, VMDBK, RDEV, VDEV, CPU, and TIME.

- You do not have to enter these values on one line (with the exception of TIME), as the prompt for selection criteria will reappear until you enter a null line or QUIT.
- You cannot specify more than a total of 16 values per selection type in all of your response lines combined, with the exception of TIME.
- An individual trace entry may or may not contain a field of the same type specified in your reply to the prompt for selection criteria. All trace entries have a trace code, CPU address, and time-of-day field; not all trace entries have a VMDBK address, real device number, or virtual device number. As a result, there are two main rules that apply to TRACERED data reduction:

- If you specify just one selection type in response to the prompt — CODE, VMDBK, RDEV, VDEV, CPU, or TIME — then the trace entries chosen for output will have a field for the selection type you indicated **and** a value in the field that matches the value you indicated in your response.

For instance, your response to the selection criteria prompt may have been the following:

```
code 0c32 1000 0100
```

In this case, all trace entries with the code 0C32, 1000, or 0100 are selected.

- If you specify more than one selection type in response to the prompt — some combination of CODE, VMDBK, RDEV, VDEV, CPU, and TIME — then the trace entries chosen for output depend on the selection type fields a given trace entry contains. Trace entries that contain fields for only some of the selection types you indicated, but whose values in those fields match the values you designated, are chosen for output. A trace entry does not have to have all the selection types you specified in order to be chosen for output.

For example, your response to the selection criteria prompt may be the following:

```
code 0c32 rdev 399 vdev 191 code 1000 0100
```

In this case, trace entries to be selected for output depend on the following:

- Trace entries with code 0C32 contain both an RDEV and VDEV value. Therefore, a trace entry of 0C32 must have an RDEV field value of 399 and a VDEV field value of 191 in order to be selected for output.
- Trace entries with code 1000 contain only an RDEV value. The VDEV selection value does not apply in this situation. Therefore, all trace entries of code 1000 with an RDEV field value of 399 are chosen for output.
- Trace entries with code 0100 have neither an RDEV nor VDEV value. Therefore, all trace entries of code 0100 are output.

8. The following guidelines apply to your responses to the prompt for selection criteria when **CP and TRSOURCE data** is requested:

- If you immediately enter a null line when prompted for the selection criteria, this indicates that no data reduction is requested. As a result, all trace entries are selected.
  - You can specify a time range only once.
9. If you own more than one set of files with different originators, you must specify NAME *fn, id* for the file you want.
  10. All files being processed must have been created under a z/VM environment.
  11. When a new system trace file or tape is being processed, a message will be issued to the screen indicating which file or tape is being processed.
  12. When you indicate VMDBK in your response, data reduction takes place by VMDBK address. Should you want to trace a specific user or users, you need to find out the VMDBK address for the user in question **before** invoking the TRACERED command.
  13. For TIME selection criteria, the handling of a two-digit year will be interpreted within a window of 100 years centered on the current date.



For example, if the year is 1997, the 100-year window spans from 1947 to 2046. Any two-digit year in the range 47-99 will be interpreted as 1947-1999, and any two-digit year in the range 00-46 will be interpreted as 2000-2046. Likewise, if the year is 2003, the 100-year window spans from 1953-2052. Any two-digit year in the range 53-99 will be interpreted as 1953-1999, and any two-digit year in the range 00-52 will be interpreted as 2000-2052.

14. When the output is directed to a CMS file, the XEDIT command will be invoked automatically. You can then scroll back and forth through the merged data.
15. When processing system trace files that contain GCS trace entries, it is advisable to use the LRECL 132 option for formatted CMS output.
16. A file of fixed length records is created when the TRSAVE subcommand extracts CP trace data. A file of variable length records is created when the TRSAVE subcommand extracts TRSOURCE trace data. These files need to remain in their original fixed or variable length format for proper input to TRACERED.
17. To further process CP records of type LAN, see [Using IPFORMAT Packet Trace Formatting Tool in z/VM: TCP/IP Diagnosis Guide](#). Note that you can convert LAN traces to the PCAP format using IPFORMAT.

## Examples

- **tracered 190 cms trace1 tables a**

TRACERED is invoked for a system trace file with spool ID 190. The user is prompted to enter selection criteria. The fully formatted output is put into CMS file TRACE1 TABLES A. XEDIT is invoked for the output file.

- **tracered 183 194 172**

TRACERED is invoked for three system trace files with spool IDs 183, 194, and 172. All defaults are taken. That is, the user is prompted to enter selection criteria, and the output is fully formatted and goes to the virtual printer.

- **tracered 0183 name pvmttrace 194 cms trace2 output a (all noxedit)**

TRACERED is invoked to merge system trace files 183 and 194 and all system trace files with file name PVMTRACE. All records are formatted and put into CMS file TRACE2 OUTPUT A. XEDIT is not invoked for the output file.

- **tracered file prb00002 trapA file prb00002 trapB A cms trace3 output a (all format)**

TRACERED is invoked to merge two CMS files created from a CP dump by the TRSAVE subcommand of DUMPSCAN. All records are formatted and put into CMS file TRACE3 OUTPUT A. XEDIT is invoked for the output file.

- **tracered name rscsprb4 orig rscsv2 ,17 cms rscs4 trace (lrecl 132)**

TRACERED is invoked for system trace files with file name RSCSPRB4 and origin ID RSCSV2, starting with spool ID 17. Output is fully formatted in 132-byte records and put in CMS file RSCS4 TRACE A.

- **tracered dev 182 name tsaf 194 (hex)**

TRACERED is invoked to merge CP trace table data on tape 182 with all system trace files with file name TSAF and with another system trace file with spool ID 194. Output is in short (hex) format and directed to the printer.

- **tracered name gcstrc orig vtam1 gtrace (all)**

TRACERED is invoked to prepare a file containing input for the ACF/TAP (Trace Analysis Program). It will contain all of the GTRACE type GCS records found in the system data files with file name GCSTRC recorded with origin ID VTAM1.

**TRACERED Output:** There are four types of output you may generate from the TRACERED command: a formatted CMS file, a formatted print file, an unformatted CMS file, or an unformatted print file.

**Note:** You may encounter storage constraints if you select a CMS file for the output. The more trace entries that meet the selection criteria, the larger the storage requirements.

One way to alleviate storage constraints is to designate more stringent selection criteria.

**Formatted CMS File:** When you select the FORMAT option (the default) and designate a CMS file in the TRACERED command, the default setting of the output to the CMS file consists of 80-character fixed-length records. You can reset the default record using the LRECL option.

**CP Trace Table Output:** Each CP trace entry selected comprises three output lines:

- The first line contains the trace code, the CPU address, a description of the type of entry, and the time the entry was created.
- The second and third lines describe the trace entry data contents, which vary according to the type of entry.

```
0600 CPU 0000 OBTAIN FREE STORAGE (FREE)      TIME 19:26:31.000000
      BLOCK ID  DOUBLE WORDS   BLOCK ADDR  VMDBK ADDR  RETURN ADDR
      <IOR      00000020      00FC5EA8   00FB1000   004CF922
```

Figure 2. Formatted CP Trace Entry

Each time TRACERED encounters a missing CP data record, the following line appears in the CMS file:

```
CPU xxxx - TRACE DATA HAS BEEN LOST
```

A blank line precedes and follows the lost data line in the CMS file. Processing continues with the next available trace entry.

Processing Input Tapes:

Each time a new input tape is processed, a header record is written that contains the volume sequence number and creation date and time of the TRSAVE tape. Also at the time a new input tape is processed, that same information is displayed at your terminal screen. Informational message O30I indicates the volume sequence number of the current TRSAVE input trace tape and the date the tape was created.

**Guest Trace Event Output:** An example of formatted GCS Event output follows:

```
3D 0E NET    VM/GCS USER REQUESTED GTRACE
      TIME OF DAY CLOCK = 9A6266195A8E2C00
      LENGTH OF GTF HEADER AND TRACE DATA =0017
      FORMAT ROUTINE ID = FD
      EVENT IDENTIFICATION = EFE1
```

Figure 3. Guest Trace Data of Type GCS in Event Mode

**Guest Trace Block Output:** An example of formatted TSAF Block output follows:

```
----- 12/15/89 14:55:20.852228 -----
10A0 GOT INTERNAL TRACE BLOCK                                TSAF
      CLOCK = A15CE563 4C704301      MODULE = ATSCTL
      TIME = 14:55:20.852228 GMT 12/15/1989  LENGTH = 0000
```

Figure 4. Formatted Guest Trace Data of Type TSAF in Block mode

**I/O Trace Entry Output:** TRACERED recognizes I/O data trace information. An example of a formatted I/O data trace entry follows:

```

TRACE TYPE IO, CPU 0000                TIME 01:00:00.000000
TRACEID - I01, TRACESET = IO, IODATA = 0
USER = OPERATOR, I/O OLD PSW = 07064000 80000000 00000000 00000000
DEVICE = 0191, SCSW = 00C04007 0FA59610 0C000000
                ESW = 00800017
I/O PRIORITIES: CHANNEL = 0, CURRENT = 0, ORIGINAL = 0
                OUT-PRIORITIZED COUNT = 0
-> CCW(1) = 63400010 0FA38510, CCW ADDRESS = 0FA38538
-> CCW(2) = 4B600016 0FA38520, CCW ADDRESS = 0FA38540
-> CCW(3) = 92400008 0FA38008, CCW ADDRESS = 0FA38548
-> CCW(4) = 86411000 0FA38290, CCW ADDRESS = 0FA38550 ** MIDA **
-> MIDAW(1) = 00000000_0E4D1000

```

Figure 5. I/O Data Formatted by TRACERED

**LAN Trace Entry Output:** TRACERED recognizes LAN data trace information.

LAN data trace information is formatted by TRACERED as:

- TOD of when entry was created.
- Contents of the trace entry in hexadecimal (up to 20 bytes per line).
- The string SPID followed by a decimal number. This identifies the system trace file from which the entry was obtained.

**Data Trace Entry Output:** TRACERED recognizes CP data trace information.

An example of a CP data trace entry output formatted by TRACERED is shown in [Figure 6 on page 2251](#).

```

TRACE TYPE DATA, CPU 0000                TIME 16:06:07.000000
TRACEID = 0, TRACESET = NULL, ADDRESS = 0061A058
-> DATALINK STRING 1 = G11.090=VMDBK
DATA = 00BDC00F 00A9E00F 00A9D00F 00A9C00F *.....Z...Z...Z.*
00A9B00F 00A9A00F 00A9900F 00A9800F *.Z...Z...Z...Z.*
00A9700F 00A9600F 00A9500F 00A9400F *.Z...Z-.z&;.z.*
00A9300F 00A9200F 00A9100F 00A9000F *.Z...&;.....*
80000020 80000020 80000020 80000020 *.....*
80000020 80000020 80000020 80000020 *.....*
80000020 80000020 80000020 80000020 *.....*
80000020 80000020 80000020 00AA0000 *.....*
D6D7C5D9 C1E3D6D9 D6D7C5D9 C1E3D6D9 *OPERATOROPERATOR*
-> DATALINK STRING 2 = G10.100=HCPTXCCM
DATA = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF *.....*
030C0000 803A30F2 000C2000 80DEB05A *.....2.....!*
000C1000 8047865C 00000000 00000000 *.....f*.....*
030C1000 804631F0 00000000 00000000 *.....0.....*
00000000 00000000 00000000 00000000 *.....*
000C0000 803B1060 000C0000 80478060 *.....-.....*
000C0000 804A0230 00080000 80401170 *.....¢.....*
000C0000 803DCCF8 0033E000 00001004 *.....8.....*
0002001C 00040016 00DE5888 00000000 *......h.....*
00000000 00000000 00000000 00000000 *.....*
00000000 00000000 00000000 00000000 *.....*
00010001 00327240 00000000 00000000 *.....*
00000000 00000000 00000000 00000000 *.....*
00000000 00000000 00000000 00000000 *.....*
00000000 00000000 00000000 00000000 *.....*
00000000 00000000 00000000 00000000 *.....*

```

Figure 6. Data Trace Entry Output Formatted by TRACERED

**Formatted Print File:** TRACERED formatted print output is identical with formatted CMS file output except that the default record length is 132 bytes. The TRACERED output is directed to your virtual printer rather than to a CMS file.

**Unformatted CMS File:** When you select the HEX option and designate a CMS file in the TRACERED command, the default setting of the output to the CMS file consists of 80-character fixed-length records. You can reset the default record using the LRECL option.

**CP Trace Table Output:** Each CP trace entry selected comprises one line. The line contains the CPU address, the time the entry was created, the trace code, and the trace entry data contents. These are in hexadecimal format.

```

CPU   SECONDS   CODE ***** TRACE ENTRY CONTENTS *****
      05/31/88 11:51:21.123456
0000  21.123456 3310  E4C34040 00000001 00FAC000 00FFBA80 005311E0 SPID 0003
    
```

Figure 7. CP Trace Table Output

**Guest Trace Block Output:** When HEX option is entered for a Block mode Guest Trace the Guest's formatting routine is invoked and its HEX output is given.

An example of HEX TSAF Block output follows:

```

----- 12/15/89 18:00:28.000000 -----
TSAF      A15C0FF4 F4BA5C01 C3E3D3 10A0 0000      GOT INTERNAL TRACE BLOCK
    
```

Figure 8. Hex Guest Trace Data of Type TSAF in Block Mode

**Other Trace Types:** TRACERED processes other traces (I/O, CP data, LAN, and guest traces when the formatting routines are not available) and produces the following information per trace record:

- TOD of when entry was created
- Contents of the trace entry in hexadecimal (up to 20 bytes per line)
- The string SPID followed by a decimal number. This identifies the system trace file from which the entry was obtained.

An example of unformatted trace data output follows:

```

----- 12/15/89 18:01:52.123456 -----
18:01:52.123456      A15C1044 BB7F8901 C3E3D310 A00000E0 0E      SPID 0086
    
```

Figure 9. Unformatted Guest Trace Output

**Unformatted Print File:** TRACERED hex print output is identical with hexadecimal CMS file output except the default record length is 132 bytes. The TRACERED output is directed to your virtual printer rather than to a CMS file.

**TRACERED Exit Descriptions:** Following are the exit descriptions.

**TRACERED Merge and Selection Exit:** TRACERED provides an interface between itself and GCS GTRACE trace data formatting routines. When you use the GCS GTRACE macro and specify the FID option (to define data to be recorded in the GCS trace table) you must provide to TRACERED a matching format routine to make your trace data suitable for printing. The name for the formatting routine must be as described in z/VM: Group Control System under the GTRACE macro.

The GTRACE macro provides you with complete control over the trace record layout, including the location of a TOD clock field. Therefore, TRACERED requires the assistance of your selection routine to obtain the TOD value for each record it processes.

**Merge/Selection Exit Interface:** When TRACERED reads a trace record in a guest trace block, it calls the exit, if it exists, to determine the TOD value or set the selection return code indicator.

*Registers at entry:*

```

R1      address of parameter list
R13     address of 72 byte save area for caller's registers
R14     return address
R15     entry point address of exit routine
    
```

*Parameter list format:*

```

x'0 '  Parameter list length
x'2 '  Flag byte
      1... .... GTRACE records requested
      .111 1111 Reserved
x'3 '  Reserved
x'4 '  Address of the trace record
x'8 '  Address of an eight byte output area for
      the TOD value supplied by the exit routine.

```

*Registers at exit:*

```

R0-R14  unchanged
R15     return code
      0      = record selected
      4      = no TOD value available
      8      = record not selected

```

The merge and selection exit returns an 8-byte time stamp for the individual record and indicates whether the record should be selected for further processing.

TRACERED selects either CP or guest trace entries by time. If the timestamp of an entry is identical to that of the previous entry, as can occur when the Store-Clock-Fast Facility is installed and enabled during tracing, an '=' sign is displayed immediately following the timestamp, signifying that the order of occurrence of those entries is unpredictable. If the timestamp of a guest record is not returned by the exit routine, the record cannot be selected by time.

The GCS merge and selection exit should select only GTRACE records when requested by TRACERED.

A return code 4 to TRACERED results in the record being skipped if the record is to be merged with others. If TRACERED is not merging multiple input files and if the records are not being selected by time, however, the record will be processed for output.

A return code of 8 indicates that the record should not be selected.

**TRACERED Formatting Exit:**

The TRACERED program supports user specified trace formatting exits for guest trace records recorded in system data files using the FID option of the GCS GTRACE macro and the TRSOURCE command (BLOCK mode).

TRACERED reads blocks of data from system trace or CMS files, separates the individual records, and passes each to the formatting exit which is identified in the block descriptor record.

Make sure that the CMS loader can access the TRACERED exit routine.

**Formatting Exit Interface:** When TRACERED processes a trace record in a system trace file block, or a CMS trace file block, it passes it to the user-supplied formatting exit if available:

*Registers at entry:*

```

R1     address of parameter list
R13    address of 72 byte save area for caller's registers
R14    return address
R15    entry point address of exit routine

```

*Parameter list format:*

```

x'0 '  Address of the trace record
x'4 '  Address of the trace record data portion
x'8 '  Length of the trace record
x'C '  Length of the trace record data portion
x'10'  Reserved, set to zero
x'14'  Address of the output buffer, into which the exit
      will build a line of formatted output.
      It is initially blank.
x'18'  Reserved, set to zero
x'1C'  Address of the output routine (in the
      TRACERED program).
x'20'  Communications Flags
      1... .... Output should be formatted in 80 byte records

```

```

                                for display. This is set when TRACERED is invoked
                                with the CMS operand, or when the output length is
                                overridden by the LRECL 80 option.
.1.. .... Output should be formatted in 132 byte records
                                for printing. This is set when TRACERED is invoked
                                with the PRINT operand, or when the output length is
                                overridden by the LRECL 132 option.
..1. .... Reserved, set to zero
...1 .... Long output format wanted. The FORMAT option
                                was specified (or defaulted) on the command.
                                This bit is zero if the HEX option was specified.
.... 1111 Reserved, set to zero

```

**Output Routine Interface:** When the user formatting exit calls the Dump Viewing Facility output routine, it will use standard linkage conventions and pass the address of the parameter list with which it was entered.

This routine uses standard exit linkage with the return code (register 15) containing an unconditional value of zero.

## Messages

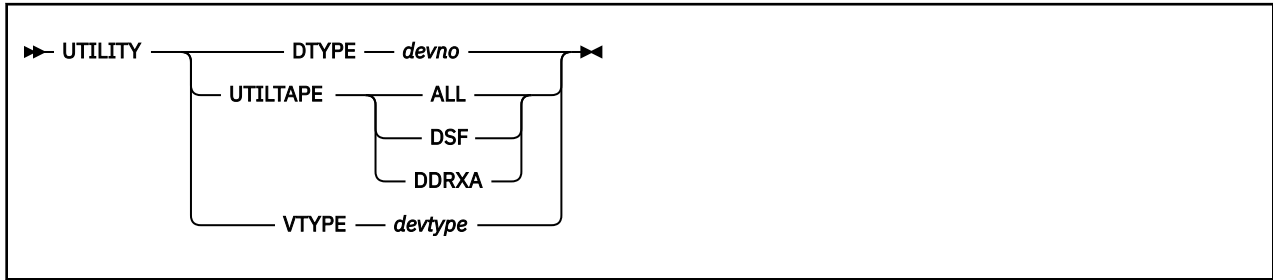
When the trace reduction process is complete, the TRACERED utility issues one of the following codes:

### **Code**

#### **Meaning**

<b>0</b>	Successful completion
<b>4</b>	No trace entries meet the selection criteria
<b>20</b>	Invalid character in output file ID
<b>24</b>	Command syntax error
<b>28</b>	Tape not attached or output file already exists or system trace file is not found
<b>36</b>	Disk not accessed
<b>41</b>	Insufficient virtual storage
<b>100</b>	I/O error

## UTILITY



### Purpose

Use the UTILITY utility to provide occasionally-used installation utility functions:

- Issuing DIAGNOSE code X'24' and X'210' for a virtual device
- Creating a stand-alone service utility tape for either or both ICKDSF and DDRXA.

### Operands

#### **DTYPE** *devno*

issues a DIAGNOSE code X'24' and X'210' for the specified virtual address, uses the returned information to look up the DASD type in the \$DASD\$ CONSTS file, and returns that information to the terminal (if UTILITY DTYPE is issued from the terminal) or to the program stack (if UTILITY DTYPE is called by an exec).

The information returned is a single line consisting of three fields:

- DASD type, model, or both
- The capacity of the device in cylinders
- The proper representation of the given DASD model to the CPFMTXA program.

If the virtual device address is incorrect, nonexistent, or addresses an unsupported device type, a nonzero return code is displayed.

The UTILITY DTYPE function is intended primarily for use by installation-related utilities.

#### **UTILTAPE**

creates a tape containing either or both of the following stand-alone utility programs:

##### **DSF:**

Device Support Facility (ICKSADSF COREIMAG)

##### **DDRXA:**

DASD Dump/Restore program (IPL DDRXA)

on a tape drive at virtual address 181.

The version of DDRXA or ICKSADSF that occurs first in the minidisk search order is the version placed on tape.

If ALL is specified, both IPLable programs are written to the same tape file, with ICKSADSF being written first. After the last program, two tape marks are written. When ALL is specified, to IPL the DDRXA program from the tape, you must IPL twice to bypass ICKSADSF.

If you do not have a R/W A-disk the tape write operation will complete but no \$MSGLOG will be created.

## UTILITY

### **VTYPE *devtype***

locates the DASD type, *devtype*, in the \$DASD\$ CONSTS file, and returns the device information to the terminal (if UTILITY VTYPE is issued from the terminal) or to the program stack (if UTILITY VTYPE is called by an exec).

The information returned is a single line consisting of three fields:

- DASD type, model, or both
- The capacity of the device in cylinders
- The proper representation of the given DASD model to the CPFMTXA program.

If *devtype* is not an acceptable device type, a nonzero return code is displayed.

The UTILITY VTYPE function is intended primarily for use by installation-related utilities.

Acceptable values of *devtype* are:

<b><u>Device Type</u></b>	<b><u>devtype</u></b>
Single-density 3380	3380
Double-density 3380	3380-E4
Triple-density 3380	3380-K4
Single-density 3390	3390-1
Double-density 3390	3390-2
Triple-density 3390	3390-3
Double-density Model-20	9336-20

### **Usage Notes**

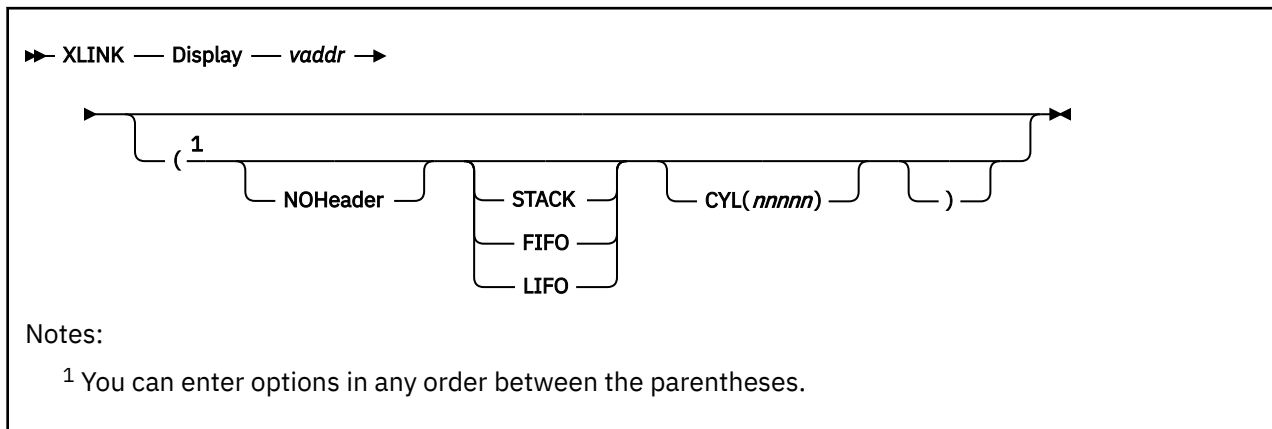
1. By default, the UTILITY utility is located on the system tools disk (MAINT 193). This utility is intended to be used only as part of a z/VM system installation procedure.

### **Messages**

- DMSOPN037E Filemode A is accessed as read/only.
- DMSEIO632E I/O error in EXECIO; rc=12 from FSOPEN command.
- HCP8300E File *filename filetype* not found.
- HCP8302E Parameter missing after *value*.
- HCP8304E Invalid parameter *parameter*.
- HCP8305E Invalid address *devno*.
- HCP8317I Moving *filename filetype* to tape.
- HCP8318I The *filename filetype* program is on tape file number *n*.



## XLINK DISPLAY



### Purpose

Use the XLINK DISPLAY utility to display link indicators in the map records of all systems on a particular device.

If a system is defined as a member of an SSI cluster, XLINK DISPLAY is not allowed.

### Operands

#### **vaddr**

specifies the virtual address of the device. This can be one of the following:

- A full-pack minidisk that defines the entire real volume
- The real volume attached to you at address *vaddr*
- A minidisk defining the cylinder on the volume that contains the CSE area.

#### **NOHeader**

keeps the descriptive column header information from being put on the CMS stack.

#### **STACK**

#### **FIFO**

#### **LIFO**

puts the information in the CMS program stack (for use by an EXEC or other program) instead of displaying it at the terminal. The information is stacked either FIFO (first in first out) or LIFO (last in first out). The options STACK and FIFO are equivalent.

#### **CYL(nnnnn)**

limits the display to the information for one minidisk starting on a specific cylinder.

### Usage Notes

1. The cross system extensions (CSE) environment is not supported. However, the cross-system link (XLINK) function that was included in CSE is still supported for non-SSI systems, and "CSE" is still used in some function and object names, command responses, and messages related to XLINK.
2. By default, the XLINK DISPLAY utility is located on the system tools disk (MAINT 193).
3. This utility requires read access to the device listed as *vaddr*.

### Responses

Response 1:

## XLINK DISPLAY

The normal response to the XLINK DISPLAY utility is:

```
HCPXXL2867I VOLSER valid NOT IN INCLUDE LIST, DEVTYPE CSE
           DEFAULTS WILL BE USED.

HCPXXL2876I VOLUME valid WAS FORMATTED FOR CSE ON
           mm/dd/yy AT hh:mm:ss
           BY USER userid AT SYSTEM sysname

           -CYL- READ/WRITE FLAGS ON valid (CYL=c TRK=t
           RECL=l RECS=r SYSTEMS=s)

           cccc   mode       sysname2   . . .   . . .   . . .
           . . .   . . .       . . .       . . .   . . .   . . .
           . . .   . . .       . . .       . . .   . . .   . . .
```

- or -

```
----- NO LINK FLAGS ON THIS VOLUME
```

### ***valid***

identifies the volume identifier of the specified device.

### ***userid***

identifies the user who formatted the CSE area on volume *valid*.

### ***sysname***

identifies the system from which the CSE area for this volume was formatted.

### ***c***

identifies the cylinder number where the CSE area was formatted.

### ***t***

identifies the track number of the CSE area.

### ***l***

identifies the number of cylinders made available for XLINK use. For CKD devices it is also the length of the XLINK map records.

### ***r***

identifies the number of XLINK records written on the CSE area.

### ***s***

identifies the number of systems that can share this DASD volume.

### ***cccc***

identifies the starting cylinder of an MDISK on volume *vvvvv* that has link flags outstanding by one or more systems.

### ***mode***

identifies the link mode, currently in effect for this MDISK by system *sysname2*. The possible mode values are:

#### **Read on**

Read-only access

#### **Write on**

Read and write access

#### **SRead on**

Read-only access and negates all write access requests.

#### **SWrite on**

Read and write access and negates all write access requests.

#### **ERead on**

Read-only access and negates all other access requests.

#### **EWrite on**

Read and write access and negates all other access requests.

**sysname2**

identifies the name of the system that has a read or write link to the MDISK that begins on cylinder *cccc*.

**Messages**

- HCP1667E THE XLINK UTILITY IS NOT ALLOWED BECAUSE THE SYSTEM IS A MEMBER OF AN SSI CLUSTER.
- HCP2861E MISSING OR INVALID FUNCTION: *subcommand*
- HCP2862E VOLSER *valid* IS NOT A CP FORMATTED VOLUME
- HCP2863E DIAG-A8 I/O ERROR, CC=*n* R15=hhhhhhh SCSW=hhhhhhh SENSE=hhhhhhh
- HCP2864E DEVICE ADDRESS *ccuu* IS INVALID OR NOT OPERATIONAL
- HCP2870E ERROR *nnn* ON CMSSTOR CALL
- HCP2871E INVALID OPTION: *option*
- HCP2872E COMMAND LINE TOO LONG FOR CP. CP CAN ACCEPT NO MORE THAN 240 CHARACTERS - YOUR COMMAND WAS *nnn* CHARACTERS IN LENGTH
- HCP2873E MISSING OR INVALID PARAMETER OR OPTION
- HCP2874E DEVICE *vdev* IS NOT A CKD OR ECKD DASD DEVICE.
- HCP2877E EITHER NOT A CSE VOLUME, OR CSE DATA IS NOT ON CYL *cccc* TRK *tt*

## XLINK FORMAT

```
▶▶ XLINK — FOrmat — vaddr — valid ▶▶
```

### Purpose

Use the XLINK FORMAT utility to prepare a DASD for sharing under cross-system link protection. The utility initializes the CSE area on the specified DASD volume.

If a system is defined as a member of an SSI cluster, XLINK FORMAT is not allowed.

### Operands

#### *vaddr*

specifies the virtual address of the device. This can be one of the following:

- A full-pack minidisk that defines the entire real volume
- The real volume attached to you at address *vaddr*
- A minidisk defining the cylinder on the volume that contains the CSE area.

#### *valid*

specifies the volume identifier that must match the one on the device.

### Usage Notes

1. The cross system extensions (CSE) environment is not supported. However, the cross-system link (XLINK) function that was included in CSE is still supported for non-SSI systems, and "CSE" is still used in some function and object names, command responses, and messages related to XLINK.
2. By default, the XLINK FORMAT utility is located on the system tools disk (MAINT 193).
3. The volume label on the disk is checked and must match the *valid* you enter with the utility.
4. You must have write access to device *vaddr*.
5. If the volume is CP-owned, you should allocate it as PERM space with the CPFMTXA utility. Even if the volume is not CP-owned, it is recommended that you allocate the CSE area as PERM.
6. If any system that will share this volume is not at release VM/ESA 1.1 or later, the CSE area must be reformatted, and the XLINK FORMAT command must be issued from a release earlier than VM/ESA 1.1. Otherwise, CP LINK commands to minidisks on this volume might fail.
7. If the number of cylinders being mapped is large or the number of tracks per cylinder is small, then XLINK FORMAT will write the ECKD format of the CSE area on multiple cylinders. The user will be given the option of stopping XLINK FORMAT before any formatting occurs on multiple cylinders, in case there are minidisks whose location needs to be adjusted.

### Responses

Response 1:

The normal responses to the XLINK FORMAT utility are:

```
HCPXXL2860I  WARNING!
             IMPROPER USE OF THE XLINK FORMAT COMMAND
             WILL CAUSE LOSS OF MINIDISK DATA. BE SURE
             THAT VOLSER valid IS NOT ON-LINE
             TO ANY ASSOCIATED SYSTEMS.

             CONTINUE?  ENTER YES OR NO:

HCPXXL2865I  VOLSER READ IS valid, FORMAT NOT DONE.
```

```
HCPXXL2867I VOLSER valid NOT IN INCLUDE LIST,
DEVTYPE CSE DEFAULTS WILL BE USED.

HCPXXL2868I XLINK FORMAT valid - CYL=cccc TRK=tt
RECL=llll RECS=rr SYSTEMS=ss

HCPXXL2896I WARNING! XLINK FORMAT COMMAND WILL
FORMAT CYLINDERS cccc-cccc

CONTINUE? ENTER YES OR NO:
```

***valid***

is the volume identifier of the specified device.

***cccc***

is the cylinder number where the CSE area is formatted.

***tt***

is the track number of the CSE area.

***llll***

is the number of cylinders made available for XLINK use. For CKD devices it is also the length of the XLINK map records.

***rr***

is the number of XLINK records written on the CSE area.

***ss***

is the number of systems that can share this DASD volume.

```
HCPXXL2869I CSE XLINK FORMAT COMPLETE.

HCPXXL2895I WARNING!
IF ALL SYSTEMS THAT WILL SHARE THIS valid
ARE NOT AT RELEASE VM/ESA 1.1, ISSUE XLINK
FORMAT FROM AN EARLIER RELEASE OF VM.

CONTINUE? ENTER YES OR NO.
```

**Messages**

- HCP1667E THE XLINK UTILITY IS NOT ALLOWED BECAUSE THE SYSTEM IS A MEMBER OF AN SSI CLUSTER.
- HCP2861E MISSING OR INVALID FUNCTION: *subcommand*
- HCP2862E VOLSER *valid* IS NOT A CP FORMATTED VOLUME
- HCP2863E DIAG-A8 I/O ERROR, CC=*n* R15=*hhhhhhh* SCSW=*hhhhhhh* SENSE=*hhhhhhh*
- HCP2864E DEVICE ADDRESS *ccuu* IS INVALID OR NOT OPERATIONAL
- HCP2870E ERROR *nnn* ON CMSSTOR CALL
- HCP2871E INVALID OPTION: *option*
- HCP2872E COMMAND LINE TOO LONG FOR CP. CP CAN ACCEPT NO MORE THAN 240 CHARACTERS - YOUR COMMAND WAS *nnn* CHARACTERS IN LENGTH
- HCP2873E MISSING OR INVALID PARAMETER OR OPTION
- HCP2874E DEVICE *vdev* IS NOT A CKD OR ECKD DASD DEVICE.
- HCP2877E EITHER NOT A CSE VOLUME, OR CSE DATA IS NOT ON CYL *cccc* TRK *tt*
- HCP2897E XLINK FORMAT CANNOT CREATE A CSE AREA STARTING AT CYLINDER *cccc* FOR *nnnn* CYLINDERS SINCE THIS IS BEYOND THE END OF THE VOLUME. XLINK FORMAT FUNCTION TERMINATED.



---

## Appendix A. Restricted User IDs

Many keywords have a specific meaning to your z/VM when they are encountered in a command input line.

The command syntax of the z/VM CP commands often lets you enter one of these keywords or a user ID with equal frequency. If you have assigned one of these keywords as a user ID, an ambiguity results that may cause the system to improperly analyze the command line.

You can avoid this problem by not using any of the following words as a user ID:

## Restricted User IDs

• *	DUMP	OBSERVER	SRM
• ABEND	DUPlex	OFF	STGEXempt
• ACCount	D8ONECMD	PAGing	STGLimit
• ALL	EXTended	PASSWORD	STORAge
• ALLDBcs	FENCES	PATHS	STore
• ALLOC	Files	PAV	SVC76
• ALLSBcs	FORCE	PCH	SWCH
• ALTID	FRames	PCIFUNCT	SWitches
• BRKKEY	Gateway	PENDING	SYSOPER
• BYUSER	GRAF	PER	SYStem
• CACHE	Hold	PFnn	S370E
• CACHEFW	HOTIO	PINNED	TAG
• CCWtran	IMG	Printer	TApes
• CFLINK	IMSG	PRIORity	TDisk
• CHANNEL	IOAssist	PRIVCLASS	TDISKCLR
• CHPID	ISLink	PROcessors	TDSK
• CMDLIMit	JOUrnal	PROcessr	TERMinal
• COLlect	LDEVs	Prt	Time
• COMMands	Links	PUnch	TIMER
• CONCOPY	LKFAC	PVMSG	TIMEZONEs
• CONsole	LINEs	QDROP	TRace
• CONV	LINEDit	QUICKDSP	TRACEFRAMEs
• COPY	LOGLxxxx	Rdr	TRFiles
• CPAssist	LOGmsg	Reader	TRSAve
• CPCHECKing	LOGNxxxx	REAL	TRSource
• CPDisks	LOGVxxxx	RECORDing	TYPe
• CPLANGLIST	MACHine	REPLY	UCR
• CPLANGuage	MAXSPOOL	REServed	UNDIAL
• CPLEVEL	MAXUsers	RESourCe	UR
• CPLOAD	MDCache	RETrieve	USERID
• CPOWNed	MDisk	RSAW	Users
• CPTRace	MITime	RUN	'V=F'
• CPUs	MOD	SASSISt	'V=R'
• CPUID	MODEL	SCReen	VDISK
• CRYPto	MONData	SDF	VDSK
• CTCa	MONitor	SEClabel	Virtual
• CTRL	MSG	SECuser	VMDump
• DAsd	MSGPROC	Set	VMSave
• DASDFW	Names	SHARE	VMSG
• DCSS	NLS	SPaces	WNG
• DISPlay	NSS	SPECIFIC	WRKALleg
	NVS	SPMODE	XSTorage
		SPOOL	XSTore



**Note:** Uppercase letters in this table represent the minimum truncations that the system will accept for any of these keywords. Where a keyword is specified in both uppercase and lowercase letters, you should not assign a user ID that is exactly the same as any acceptable truncation of the keyword. For example:

STORage

is a reserved keyword. You should not assign any of the following as user IDs:

STOR  
STORA  
STORAG  
STORAGE

Users should not be given user IDs that match SNA device LUNAMES.

Userids starting with LOGNxxxx, LOGLxxxx, and LOGVxxxx are used by the system and are restricted. The xxxx can be any character, number, or symbol.

**Note:** A one-to-four digit spoolid may be treated as a userid if it is in the directory.

The characters that you may use for user IDs specified in system configuration file statements are restricted by the USERID\_CHARACTERS and by the USERID\_PATTERN translate test tables. These tables restrict legal user ID characters to these:

**alphabetic**

A through Z

**numerics**

0 through 9

**others**

@ # \$ \_ (underscore) - (hyphen)

It is recommended that you not use logical line editing symbols in userids. See [Appendix B, “Logical Line Editing Symbols,”](#) on page 2267.

If you must change these legal character restrictions, please refer to [CHARACTER\\_DEFAULTS Statement](#) in *z/VM: CP Planning and Administration*.



## Appendix B. Logical Line Editing Symbols

To aid you in entering command or data lines from your line mode or display terminal, z/VM provides a set of logical line editing symbols, which you can use to correct mistakes as you enter lines. Each symbol has been assigned a default character value. These normally are:

<b>Symbol</b>	<b>Character</b>
Logical character delete	@
Logical line end	#
Logical line delete	␣
Logical escape	"

### Logical Character Delete

The logical character delete symbol (@) lets you delete one or more of the previous characters entered. The @ deletes one character per @ entered, including the ␣ and # logical editing characters. For example,

```
ABC#@@    results in AB
ABC@D    results in ABD
␣@DEF    results in DEF
ABC@@@   deletes the entire string
```

### Logical Line End

The logical line end symbol (#) lets you enter more than one command on the same line, and thus minimizes the amount of time you have to wait between entering commands. You type the # at the end of each logical command line, and follow it with the next logical command line. z/VM stacks the commands and runs them in sequence. For example, the entry

```
query users#query time#query dasd
```

is run in the same way as the entries

```
query users
query time
query dasd
```

The logical line end symbol also has special significance for the #CP function. Beginning any physical line with #CP indicates that you are entering a command that is to be processed by z/VM immediately. If you have set a character other than # as your logical line end symbol, you should use that character instead of a #.

### Logical Line Delete

The logical line delete symbol (␣) deletes the entire previous physical line, or the last logical line back to (and including) the previous logical line end (#). You can use it to cancel a line containing many or serious errors. If a # immediately precedes the ␣ sign, only the # sign is deleted, because the # indicates the beginning of a new line, and the ␣ cancels the current line. For example:

- Logical Line Delete:

```
ABC#DEF␣    deletes the #DEF and results in ABC
ABC#␣       results in ABC
ABC#DEF␣#GHI results in ABC#GHI
ABC#DEF␣GHI results in ABCGHI
```

## Logical Line Editing Symbols

- Physical Line Delete:

```
ABC␣          deletes the whole line
```

**Note:** When you cancel a line by using the ␣ logical line delete symbol, you do not need to press a carriage return; you can continue entering data on the same line.

## Logical Escape

The logical escape symbol (") causes z/VM to consider the next character entered to be a data character, even if it is normally one of the logical line editing symbols (@, ␣, ", or #). For example,

```
ABC"␣D        results in ABC␣D
"ABC"         results in "ABC"
```

If you enter a single logical escape symbol (") as the last character on a line, or on a line by itself, it is ignored.

When you enter multiple logical escape characters in conjunction with other logical editing characters, the results may be difficult to predict. For example, the lines

```
ABC""@DEF
ABC""@@DEF
```

both result in the line ABCDEF.

## Defining Logical Line Editing Symbols

The logical line editing symbols are defined for each virtual machine in the virtual machine's CP directory entry. If your terminal's keyboard lacks any of these special characters, your installation can define other special characters for logical line editing. To find out the logical line editing symbols in effect for your virtual machine, enter the command

```
cp query terminal
```

The response might be

```
LINEND # , LINEDEL ␣ , CHARDEL @ , ESCAPE " , TABCHAR ON
LINESIZE 080 , ATTN OFF , APL ON , TEXT OFF , MODE VM , HILIGHT ON
CONMODE 3215 , BREAKIN IMMED , BRKKEY PA1 , SCRNSAVE OFF
AUTO CR ON
```

You can use the TERMINAL command to change the logical line editing characters for your virtual machine. For example, if you enter

```
terminal linend /
```

then, the line

```
input # line / input / #
```

is interpreted

```
input # line
input
#
```

The terminal characteristics listed in the response to the QUERY TERMINAL command are all controlled by operands of the TERMINAL command.

## Appendix C. ASCII Translation

z/VM supports ASCII devices using 37x5. This section describes enhancements to VM support of ASCII in connection with line mode support.

### Provision of Translate Tables

The `TERMINAL` command has as an operand, `ASCIITBL`, that can be used on TTY type terminals. `ASCIITBL` lets you select the translation tables you want to use from the following:

#### VM1

ASCII translate tables based on STD TTY ANSI X3.4 1977 translate tables. VM1 is the default.

#### VM2

ASCII translate tables based on STD TTY ANSI X3.26 1980 translate tables.

If you find translate table discrepancies because your application uses STD TTY ANSI X3.26 1980 translate tables, you can select these translate tables with the VM2 option. If not, you can continue to use the 1977 translate tables as the default or by selecting the VM1 option.

### ASCII to EBCDIC Differences

ASCII	VM1 EBCDIC	Graphic	VM2 EBCDIC	Graphic
21	5A	!	4F	
5B	AD	[	4A	¢
5D	BD	]	5A	!
7C	4F		6A	!

### EBCDIC to ASCII Differences

ASCII	VM1 EBCDIC	Graphic	VM2 EBCDIC	Graphic
4A	7F	DEL	5B	[
4F	7C		21	!
5A	21	!	5D	]



## Appendix D. HCPLDR Control Statements and Output

The section provides information about the loader control statements and the HCPLDR output for future diagnostic purposes.

### Loader Control Statements

The following general rules apply to loader control statements:

1. The first column of the loader control statement must be X'02'.
2. Columns 2–4 must contain the statement name.
3. Unless stated otherwise:
  - operands can appear anywhere between columns 5 and 72
  - operands must consist of arguments of no more than 8 characters.
4. Operands must be separated by blanks.

### Printer Control Statement

The printer control statement enables or disables printing of the cross-reference listings and load maps.

Format:

Column 1	Columns 2-4	Columns 5-72
X'02'	PRT	{ON   <i>rdevno</i>   OFF }

Where:

**X'02'**

must appear in column one.

**PRT**

is the printer control statement acronym.

**{ON | *rdevno* | OFF }**

ON and *rdevno* enable printing of cross-reference listings and the load map. If you specify ON, the printer is at 000E. If you specify *rdevno*, the printer is at that device number.

OFF disables printing. If you enter the HCPLDR command without the PUNCH or TAPE option, this operand is ignored.

**Note:** Only one operand may appear on the statement.

### Set Page Boundary Control Statement

The set page boundary control statement aligns the next CSECT according to the page size specified.

Format:

Column 1	Columns 2-4	Columns 5-72
X'02'	SPB	<i>nnnnn</i>

Where:

**X'02'**

must appear in column 1.

### SPB

is the set page boundary control statement acronym.

### *nnnnn*

is the page size in hexadecimal. The page size may be any power of 2 from 8 to 65536. If not specified, the page size is X'1000' (decimal 4096).

**Note:** Only one argument may appear on the statement.

## Unconditional Page Boundary Control Statement

The unconditional page boundary control statement aligns all subsequent CSECTs according to the page size specified.

Format:

Column 1	Columns 2-4	Columns 5-72
X'02'	UPB	<i>nnnnn</i>

Where:

### **X'02'**

must appear in column 1.

### **UPB**

is the unconditional page boundary control statement acronym.

### *nnnnn*

is the page size in hexadecimal. The page size may be any power of 2 from 8 to 65536. The initial value is 8. If not specified, the page size is X'1000' (decimal 4096).

**Note:** Only one argument may appear on the statement.

## Conditional Page Boundary Control Statement

The conditional page boundary control statement conditionally aligns all subsequent CSECTs according to the page size specified. If a CSECT will not fit within the remainder of the present page, the system loader will locate the CSECT on the next page boundary.

Format:

Column 1	Columns 2-4	Columns 5-72
X'02'	CPB	<i>nnnnn</i>

Where:

### **X'02'**

must appear in column 1.

### **CPB**

is the conditional page boundary control statement acronym.

### *nnnnn*

is the page size in hexadecimal. The page size may be any power of 2 from 8 to 65536. If not specified, the page size is X'1000' (decimal 4096).

**Note:** Only one argument may appear on the statement.

Usage Notes®:

1. The standard value for page sizes is X'1000' (decimal 4096).

## Set Location Counter Control Statement

The set location counter control statement sets the next CSECT at an absolute address.



Format:

Column 1	Columns 2-4	Columns 5-23	Columns 24-80
X'02'	SLC	<i>nnnnnnnn</i>	ignored

Where:

**X'02'**

must appear in column 1.

**SLC**

is the set location counter acronym.

***nnnnnnnn***

is the absolute address in hexadecimal.

**Note:** Only one argument may appear on the statement.

Usage Notes:

1. Module HCPSYS uses the set location counter control statement to set the base address of the CP nucleus.
2. CMS uses the set location counter control statement to set the transient and user areas.

## Padding Control Statement

The padding control statement fills all unspecified locations in a CSECT with a specified value.

Format:

Column 1	Columns 2-4	Columns 5-72
X'02'	PAD	<i>nn</i>

Where:

**X'02'**

must appear in column 1.

**PAD**

is the padding control statement acronym.

***nn***

is the fill character in hexadecimal. The initial value is 0.

**Note:** Only one argument may appear on the statement.

Usage Notes:

1. Padding helps in debugging a program. If a program references unspecified areas, you can detect such references by padding with a value of X'F0' or some similar value.

## Parameter Control Statement

The parameter control statement supplies a parameter string that a loaded system uses upon execution.

Format:

Column 1	Columns 2-4	Columns 5-72
X'02'	PRM	<i>parmstring</i>

Where:

**X'02'**

must appear in column 1.

### PRM

is the parameter control statement acronym.

### *parmstring*

is a parameter string having a 2-byte length field followed by the parameter. When the system begins execution, general purpose register one addresses this parameter string.

## Subsystem Control Statement

The subsystem control statement specifies a subsystem and the location within a virtual machine where the subsystem appears. Such a subsystem is a separately loaded system, having its own relocation, page boundaries, cross reference listings, and load maps. However, the system loader loads the subsystem as a part of the main system. Use such a "system loaded within a system" for stand-alone programs. You can load the programs as part of the CP nucleus.

Format:

Column 1	Columns 2-4	Columns 5-72
X'02'	SYS	<i>name</i> [ <i>loc</i> ]

Where:

### **X'02'**

must appear in column 1.

### **SYS**

is the subsystem control statement acronym.

### *name*

is the name of the subsystem.

### *loc*

is the location where the subsystem appears in the virtual machine. If omitted, the location is 0.

## Loader Termination Control Statement

The loader termination control statement terminates the current system or subsystem. Control is passed to the entry point specified on the loader termination control statement. Every subsystem control statement must have its corresponding loader termination control statement and there must be one loader termination statement for the base system.

Format:

Column 1	Columns 2-4	Columns 5-72
X'02'	LDT	<i>name</i>

Where:

### **X'02'**

must appear in column 1.

### **LDT**

is the loader termination control statement acronym.

### *name*

is the entry point name that receives control after the system nucleus is loaded into storage. This operand is ignored for subsystems.

## Replace Control Statement

The replace control statement replaces instructions and constants in virtual storage.

Format:

Column 1	Columns 2-4	Columns 5-6	Columns 7-12	Columns 13-14	Columns 15-16	Columns 17-70	Columns 71-72	Columns 73-80
X'02'	REP		<i>sadd</i>		<i>esdid</i>	<i>flds</i>		nu

Where:

**X'02'**

must appear in column 1.

**REP**

is the replace control statement acronym.

***sadd***

is the hexadecimal starting address of the area that will be replaced.

***esdid***

is the External Symbol Identification, a number assigned by the compiler or assembler to the CSECT in which the replacement occurs.

***flds***

can be up to eleven 4-digit hexadecimal fields, separated by commas, each replacing one halfword. A comma cannot follow the last field.

**nu**

means "not used" by the loader. The field may be left blank or contain program identification.

Usage Notes:

1. The replace control statement must be in hexadecimal code.
2. The data in columns 17–70 (excluding commas) on the replace statement replaces what already was loaded into virtual storage beginning at the address specified in columns 7–12.
3. A replace statement may appear immediately preceding the:
  - END statement of an object module, if the module does not contain relocatable data (such as address constants)
  - First relocatable dictionary statement, if the object module does contain relocatable data.
4. If additions made by the replace statements increase the length of a CSECT, an include control section control statement must appear preceding the object module.
5. If the most recent VER control statement for a CSECT failed, the replace operation for that CSECT does not occur.

## Verify Control Statement

The verify control statement verifies a corresponding replace control statement, that is, verify assures that the proper data is replaced.

Format:

Column 1	Columns 2-4	Columns 5-6	Columns 7-12	Columns 13-14	Columns 15-16	Columns 17-70	Columns 71-72	Columns 73-80
X'02'	VER		<i>sadd</i>		<i>esdid</i>	<i>flds</i>		nu

Where:

**X'02'**

must appear in column 1.

**VER**

is the verify control statement acronym.

***sadd***

is the hexadecimal starting address of the area to be verified.

## ***esdid***

is the External Symbol Identification, a number assigned by the compiler or assembler to the CSECT in which the verification occurs.

## ***flds***

is up to eleven 4-digit hexadecimal fields, separated by commas, each verifying previously loaded halfwords. A comma cannot follow the last field.

## ***nu***

means "not used" by the loader. This field may be blank or contain program identification.

Usage Notes:

1. The verify control statement must be in hexadecimal code.
2. The data in columns 17–70 (excluding commas) on the verify statement verifies what already was loaded into virtual storage beginning at the address specified in columns 7–12.
3. Place verify control statements immediately before corresponding replace control statements.
4. If the verify function fails:
  - The loaded system does not run.
  - All following REP control statements for that CSECT are ignored until the next successful VER control statement for that CSECT.

## **Include Control Section Control Statement**

The include control section statement changes the length of a CSECT. Use it only when replace control statements cause a control section to increase in length.

Format:

<b>Column 1</b>	<b>Columns 2-4</b>	<b>Columns 5-72</b>
X'02'	ICS	<i>name size</i>

Where:

### **X'02'**

must appear in column 1.

### **ICS**

is the include control section statement acronym.

### ***name***

is the name of the CSECT you wish to enlarge.

### ***size***

is the total size required for the CSECT. If size is smaller than the assembled size, the assembled size is used. The value is hexadecimal.

Usage Notes:

1. Both arguments in the include control section statement are required.
2. The include control section statement must appear before the first ESD statement of the CSECT.

## **Delete Control Statement**

The delete control statement deletes up to eight CSECTs.

Format:

<b>Column 1</b>	<b>Columns 2-4</b>	<b>Columns 5-72</b>
X'02'	DEL	<i>name1 [... name8]</i>

Where:

**X'02'**

must appear in column 1.

**DEL**

is the delete control section statement acronym.

**name1 [...name8]**

is up to eight CSECT names. These names must not be previously defined within the load. Subsequent attempts to define these names as part of the system or subsystem will result in those CSECTs being deleted.

## Output

HCPLDR output for the MAP option consists of a file that contains several cross-reference listings and load maps. These listings and load maps from HCPLDR contain information provided for future diagnostic purposes. HCPLDR output for the SMALLMAP option consists of a file that contains CSECT information. This information is provided for future diagnostic purposes.

## Text Deck Information

This section contains the following information about each text deck (object file) in the generated system:

- The name of the text deck and its history
- The CSECT name, location, and size
- The names of the entry points and their locations
- Any loader control statements included in the text deck.

A one record header precedes the text deck information section. This header prevents the loss of information when HCPLDR output is PEEKed or RECEIVED by CMS. The header card produced by HCPLDR looks like this:

```
:READ  HEADER  HCPLDR  A1 AAAAAA 00/00/00  00:00 PEEK/RECEIVE REMOVE THIS CARD
```

The format of the header card is defined in the description of the CMS PUNCH command. For more information, see [PUNCH command](#) in *z/VM: CMS Commands and Utilities Reference*.

## System Summary

This section shows the labels that mark the beginning and end of the system, and its size in hexadecimal. @MAPSTRT and @MAPEND are the beginning and ending addresses of the load map. The module size is the size of the CP load map in bytes. @LOADEND is the next highest page address after the end of the map. [Figure 10 on page 2277](#) shows an example of this information from a CP load map.

```
@MAPSTRT AT 004A7000  MODULE SIZE IS 013A60
@MAPEND  AT 004BAA60
@LOADEND AT 004BB000  MODULE SIZE IS 000000
```

*Figure 10. System Summary for CP*

## CSECTs with Size Greater than Conditional Page Boundary

This report lists the pageable modules whose size is greater than the conditional page boundary setting.

[Figure 11 on page 2278](#) shows an example of this report from a CP load map. Any modules other than @MAPSTART that appear in this report are in error.

```
CSECT'S WITH SIZE GREATER THAN CONDITIONAL PAGE BOUNDARY
@MAPSTART 026D00
```

Figure 11. CSECTs with Size Greater than Conditional Page Boundary

## External Cross Reference: Symbol to Adcon CSECT

This report shows the modules that reference each entry point.

Listed by entry point, this report shows the offset within the module at which the entry point is referenced and the length of the external reference. [Figure 12 on page 2278](#) shows an excerpt from this report from a CP load map.

```
EXTERNAL CROSS REFERENCE: SYMBOL TO ADCON CSECT, ASSEMBLED LOCATION AND LENGTH
HPCFCFMS AT HPCFCFF 000100 AL4 HPCFCMD 000238 AL4 HPCFCNS 001938 AL4 HPCFCRF 0007E4 AL4
HPCPLGO 000854 AL4 HPCFCLOG 000A84 AL4 HPCFCSECCR 0000F0 AL4 HPCFCVCP 000A8C AL4
HPCFCVYRD 000910 AL4
HPCFCFMRD AT HPCFAUT 0007AC AL4 HPCFCPLD 000560 AL4 HPCFCNS 001944 AL4 HPCFCEXT 0002DC AL4
HPCFCGRF 0007E0 AL4 HPCFCGRF 001884 AL4 HPCFCGRL 000240 AL4 HPCFCIOA 001680 AL4
HPCFCPIUS 000248 AL4 HPCFCLOG 000A34 AL4 HPCFCPLPF 000EF4 AL4 HPCFCPLPN 000F5C AL4
HPCFCPTR 001784 AL4 HPCFCQCN 000AFC AL4 HPCFCQCS 000500 AL4 HPCFCSTP 000E8C AL4
HPCFCVCN 000DF0 AL4 HPCFCVCN 000E2C AL4 HPCFCVCP 000A7C AL4 HPCFCVCU 000BB4 AL4
HPCFCVOL 000A0C AL4 HPCFCVRV 0001EC AL4
```

Figure 12. External Cross Reference

## Cross Reference: CSECT to Referenced Symbol

This report shows the entry points that each module references.

Listed by module, this report shows the offset within the module at which the entry point is referenced and the length of the external reference.

[Figure 13 on page 2278](#) shows an excerpt from this report from a CP load map.

A hyphen after the length indicates that the value of the external was subtracted.

```
CROSS REFERENCE: CSECT TO REFERENCED SYMBOL, ASSEMBLED ADCON LOCATION AND LENGTH
HPCFCFM TO HPCFALLC 001030 AL4 HPCFALLR 001034 AL4 HPCFCFFBG 0010C0 AL4 HPCFCFMBK 001098 AL4
HPCFCMDRT 00107C AL4 HPCFCMDWO 001064 AL4 HPCFCDSBOW 00104C AL4 HPCFCDSBSW 00105C AL4
HPCFCIOACF 001044 AL4 HPCFCIOACX 001080 AL4 HPCFCIOAIT 001090 AL4 HPCFCIOARC 00108C AL4
HPCFCIOLAV 001038 AL4 HPCFCIOLDV 0010B8 AL4 HPCFCIOLGV 0010B4 AL4 HPCFCIOLRR 001054 AL4
HPCFCPLCKRS 001040 AL4 HPCFCPTRMS 001068 AL4 HPCFCQCNRD 0010AC AL4 HPCFCPREIPL 001084 AL4
HPCFCSTKCF 0010B0 AL4 HPCFCSTKDT 001060 AL4 HPCFC SVC 00072A AL4- HPCFC SVCCS 00072A AL4
HPCFC SVCR 001070 AL4 HPCFC TRDSP 00106C AL4 HPCFC TRQRT 001048 AL4 HPCFC TRQST 0010A4 AL4
HPCFCVCOAT 0010C4 AL4
HPCFCFO TO HPCFCFLLVM 0004B0 AL4 HPCFCFOQQ 0004A8 AL4 HPCFCVTBD 0004B4 AL4 HPCFCVTDB 00049C AL4
HPCFCGVSVD 0004CC AL4 HPCFCMCMOD 0004C8 AL4 HPCFCSCVMD 000494 AL4 HPCFC SVC 000044 AL4-
HPCFC SVC 000092 AL4- HPCFC SVC 0000CA AL4- HPCFC SVC 000114 AL4- HPCFC SVC 000128 AL4-
HPCFC SVC 000330 AL4- HPCFC SVC 0003EC AL4- HPCFC SVCCS 000044 AL4 HPCFC SVCCS 00006C AL4
HPCFC SVCCS 0000CA AL4 HPCFC SVCCS 000114 AL4 HPCFC SVCCS 000128 AL4 HPCFC SVCCS 00015C AL4
HPCFC SVCCS 0003EC AL4
```

Figure 13. CSECT to Referenced Symbol Report

## CSECT Cross Reference: Referenced CSECT to Adcon Location

This report shows the modules that reference each module.

[Figure 14 on page 2279](#) shows an excerpt from this report from a CP load map.

```

CSECT CROSS REFERENCE: REFERENCED CSECT TO ADCON LOCATION
HCPCFM AT HCPAUT HCPCFF HCPCLD HCPCMC HCPCMD HCPCNF HCPCNS HCPDIA HCPEXT HCPGFS
          HCPGRL HCPIOA HCPIUS HCPLGO HCPLOG HCPLPF HCPLPN HCPPFX HCPPTR HCPQCN
          HCPSTP HCPTRC HCPTRD HCPUSO HCPVCN HCPVCP HCPVCU HCPVCX HCPVCYRD HCPVOL
HCPCFO AT HCPQUY HCPSET
HCPCFS AT HCPSET
HCPCFT AT HPCCOM
    
```

Figure 14. Referenced CSECT to Adcon

### CSECT Cross Reference: Adcon Location to Referenced CSECT

This report shows the modules that are referenced by each module.

Figure 15 on page 2279 shows an excerpt from this report from a CP load map.

```

CSECT CROSS REFERENCE: ADCON LOCATION TO REFERENCED CSECT
HCPCFF TO HCPCFM
HCPCFM TO HCPALL HCPCFF HCPCFMBK HCPCFMHX HCPCMD HCPDSB HCPGFS HCPIOA HCPVOL HCPLCK
          HCPREIPL HCPSECAD HCPSTK HCPSVC HCPTRD HCPTRQ HCPUSP HCPVCO
HCPCFO TO HCPALL HCPCFOQQ HCPCVT HCPDSB HCPGSV HCPMCM HCPSCV HCPSVC
HCPCFS TO HCPASI HCPNCOF HCPNCON HCPGSV HCPIOA HCPRST HCPSVC
    
```

Figure 15. Adcon Location to Referenced CSECT

### Assembly Dates and Entry Points

Listed by module, this report shows the entry points in each module and the time and day each module was assembled.

Figure 16 on page 2279 shows an excerpt from this report from a CP load map.

```

CSECT      ASSEMBLY DATES AND ENTRY POINTS
           ASM DATE DAY  ENTRY POINTS
-----
HCPCFC     4/12/90 THU
HCPCFCMD   4/12/90 THU
HCPCFCSD   4/12/90 THU
HCPCFD     4/12/90 THU  HCPCFDLO HCPCFDSY
HCPCFF     4/12/90 THU  HCPCFFBG
HCPCFM     4/10/90 TUE  HCPCFMAL HCPCFMBK HCPCFMFO HCPCFMHX HCPCFMKC HCPCF MKS HCPCFMRC HCPCFMRD
HCPCFO     4/12/90 THU  HCPCFOMD HCPCFOQQ HCPCFOQR HCPCFORS
HCPCFS     4/12/90 THU  HCPCFSCN HCPCFSCP HCPCFSCT HCPCFSCW HCPCFSEM HCPCF SIM HCPCFSLE HCPCFSMA
           HCPCFSSM HCPCFSSV HCPCFSTI HCPCFSVM HCPCFSWN
    
```

Figure 16. Assembly Dates and Entry Points

### CSECT Assembly Dates, Least to Most Recent

Sorted by date (oldest to most recent), this report shows the time and day each module was assembled.

Figure 17 on page 2280 shows an excerpt from this report from a CP load map.

## HCPLDR Control Statements and Output

CSECT	CSECT DATE	ASSEMBLY CSECT	DATES, DATE	LEAST TO CSECT	MOST RECENT DATE	CSECT	DATE	CSECT	DATE
HCPURPUN	4/07/90	HCPDEO	4/10/90	HCPHVR	4/10/90	HCPMNBUR	4/10/90	HCPMXNSC	4/10/90
HCPVMI	"	HCPDGG	"	HCPIIR	"	HCPMMEI	"	HCPMXS	"
HCPVSR	"	HCPDGR	"	HCPIMSTD	"	HCPMNT	"	HCPMXSPU	"
HCPXIOCP	"	HCPDMD	"	HCPINS	"	HCPMNTBB	"	HCPMXSRS	"
HCPABF	4/10/90	HCPDMW	"	HCPINSCW	"	HCPMNTCA	"	HCPMXSSP	"
HCPALL	"	HCPDPS	"	HCPINSTC	"	HCPMNTCD	"	HCPMXVAL	"
HCPCCW	"	HCPDVT	"	HCPIISC	"	HCPMONBU	"	HCPNSN	"
HCPCCWTR	"	HCPERA	"	HCPIISH	"	HCPMONFF	"	HCPNSO	"
HCPCFM	"	HCPERM	"	HCPIISR	"	HCPMOO	"	HCPNSP	"
HCPCHM	"	HCPERBM	"	HCPIITM	"	HCPMOOHF	"	HCPNSR	"

Figure 17. Assembly Dates, Least to Most Recent

## Load Map Sorted by Name

This report shows each module and entry point, sorted by name.

Figure 18 on page 2280 shows an excerpt from this report from a CP load map.

LOAD MAP SORTED BY NAME									
NAME	LOAD	ASM	SIZE	NAME	LOAD	ASM	SIZE		
HCPCFM	SD	002678A0	000000	001108	HCPCHRLC	00268BC0	000120		
HCPCFMAL		00268890	000FF0		HCPCHRLD	00268E20	000380		
HCPCFMBK		002687C0	000F20		HCPCHRLF	00268CF0	000250		
HCPCFMFO		00268760	000EC0		HCPCHRLH	00268F50	0004B0		
HCPCFMHX		002687D0	000F30		HCPCHS	SD	00269010	000000	000000
HCPCFMKC	--	00268820	000F80		HCPCHSTR	SD	00269010	000000	000108
HCPCFMKS		002687E0	000F40		HCPCHSUN	SD	00269120	000000	0000B8
HCPCFMRC	--	002685C8	000D28		HCPCHIN	SD	002691E0	000000	000250
HCPCFMRD		002678C8	000028		HCPCHINVA		00269208	000028	
HCPCFMTL	--	00268718	000E78		HCPCHIO	SD	00269430	000000	000000
HCPCFMWC	--	002686B0	000E10		HCPCHIOCS	SD	00269430	000000	000108
HCPCFMWU		00268680	000DE0		HCPCHIOOI	SD	00269540	000000	0001B0
HCPCFMXT		00268860	000FC0		HCPCKP	SD	0024A000	000000	000A20
HCPCF0	SD	00447570	000000	000510	HCPCKPCK		0024A028	000028	
HCPCFOMD		00447870	000300		HCPCKPRS		0024A088	000088	
HCPCF0QQ		00447788	000218		HCPCKPSH		0024A030	000030	
HCPCF0QR		00447678	000108		HCPCKS	SD	0024B000	000000	000A70
HCPCFORS		00447598	000028		HCPCKSDQ		0024B168	000168	
:									
HCPIMSQY		002A55EE	00062E		HCPCHIOB	SD	002A9A90	000000	0004B0
HCPIMSTD	SD	002A4FC0	000000	0006F8	HCPCHIOBAP	..	002A9DE0	000350	
HCPIMSTO		002A5544	000584		HCPCHIOBCA		002A9AB8	000028	

Figure 18. Load Map Sorted by Name

Table 51 on page 2280 shows the contents of the report for modules and entry points.

Table 51. Load Map Contents		
Label	Contents for Modules (CSECTs)	Contents for Entry Points
NAME	The name of the module	The name of the entry point



*Table 51. Load Map Contents (continued)*

Label	Contents for Modules (CSECTs)	Contents for Entry Points
(Blank)	SD	A reference code: -- The entry point is unreferenced. ** The entry point is referenced by an EXTRN but not by any address constant. <b>(blank)</b> The entry point is referenced by an EXTRN with an address constant (most CP entry points fall into this category).
LOAD	The load address of the module	The load address of the entry point
ASM	The offset of the module name within the module (for CP, this is usually 0)	The offset of the entry point within the module
SIZE	The size of the module	(Blank)
(Blank)	Normally blank. ? if an attempt was made to define the symbol twice.	Normally blank. ? if an attempt was made to define the symbol twice.

### Load Map Sorted by Address

This report contains the same information as the Load Map Sorted by Name, but it is sorted by the load address of the module or entry point.

You can use this report to check the starting and ending addresses of your generated system and verify that modules were loaded in the correct order.

### Other Information

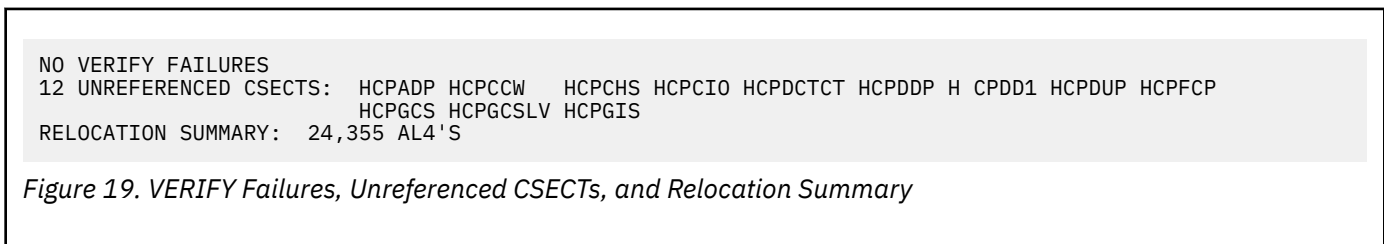
Figure 19 on page 2281 shows an example of the last report from a CP load map. It shows:

- The number of failures of any VERIFY loader control statements contained in the system.

Your system should not have any verify failures. Since VERIFY and REPLACE loader control statements are most often used to replace object code in a text deck, a verify failure probably indicates an error in the resulting code.

- The number of unreferenced CSECTs and their names.
- The relocation summary. The relocation summary shows the number of references by length (1, 2, 3, and 4 bytes).

For CP, all of the references are 4 bytes long (AL4s).



## SMALLMAP File Information

If the SMALLMAP option is used, HCPLDR will produce a file with the following characteristics. The example given is for a CP system and [Figure 20 on page 2282](#) shows an excerpt of its contents.

Filename	Filetype	Fm	Format	Lrecl	Records	Blocks	Date	Time
CPLoad	SMALLMAP	A1	V	39	1667	34	2/17/93	17:30:21
HCPABN	00032260	000CA8	01/29/93	20:22				
HCPABS	00032F10	000420	01/29/93	20:22				
HCPABW	00034000	009760	01/23/93	20:02				
HCPADP	0003D760	000000	01/29/93	21:31				
HCPADPTR	0003D760	000100	01/29/93	21:31				
HCPAEP	0003D860	000000	01/29/93	21:42				
HCPAEPKA	0003D860	0006E0	01/29/93	21:42				
HCPAEPKI	0003DF40	000140	01/29/93	21:42				
HCPAEPDA	0003E080	0001B8	01/29/93	21:42				
HCPAEPPI	0003E240	0001A8	01/29/93	21:42				
HCPAEPQC	0003E3F0	000160	01/29/93	21:42				
HCPAEPQV	0003E550	0001A0	01/29/93	21:42				
HCPAEPRI	0003E6F0	0007B0	01/29/93	21:42				
HCPAER	0003EEA0	000298	01/29/93	20:22				
HCPAFF	0003F140	000B68	01/29/93	20:22				

*Figure 20. Contents of SMALLMAP option*

- first token = csect name
- second token = address
- third token = csect size
- fourth token = text file date
- fifth token = text file time

**Note:** Date/time refer to the date/time of the text file from which the csect was read.

## Notices

---

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

#### COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

## Trademarks

---

IBM, the IBM logo, and [ibm.com](http://ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on [IBM Copyright and trademark information](https://www.ibm.com/legal/copytrade) (<https://www.ibm.com/legal/copytrade>).

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

## Terms and Conditions for Product Documentation

---

Permissions for the use of these publications are granted subject to the following terms and conditions.

### Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

### Personal Use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

## Commercial Use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

## Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## IBM Online Privacy Statement

---

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see:

- The section entitled **IBM Websites** at [IBM Privacy Statement](https://www.ibm.com/privacy) (<https://www.ibm.com/privacy>)
- [Cookies and Similar Technologies](https://www.ibm.com/privacy#Cookies_and_Similar_Technologies) ([https://www.ibm.com/privacy#Cookies\\_and\\_Similar\\_Technologies](https://www.ibm.com/privacy#Cookies_and_Similar_Technologies))



# Bibliography

---

This topic lists the publications in the z/VM library. For abstracts of the z/VM publications, see [z/VM: General Information](#).

## Where to Get z/VM Information

---

The current z/VM product documentation is available in [IBM Documentation - z/VM \(https://www.ibm.com/docs/en/zvm\)](https://www.ibm.com/docs/en/zvm).

## z/VM Base Library

---

### Overview

- [z/VM: License Information](#), GI13-4377
- [z/VM: General Information](#), GC24-6286

### Installation, Migration, and Service

- [z/VM: Installation Guide](#), GC24-6292
- [z/VM: Migration Guide](#), GC24-6294
- [z/VM: Service Guide](#), GC24-6325
- [z/VM: VMSES/E Introduction and Reference](#), GC24-6336

### Planning and Administration

- [z/VM: CMS File Pool Planning, Administration, and Operation](#), SC24-6261
- [z/VM: CMS Planning and Administration](#), SC24-6264
- [z/VM: Connectivity](#), SC24-6267
- [z/VM: CP Planning and Administration](#), SC24-6271
- [z/VM: Getting Started with Linux on IBM Z](#), SC24-6287
- [z/VM: Group Control System](#), SC24-6289
- [z/VM: I/O Configuration](#), SC24-6291
- [z/VM: Running Guest Operating Systems](#), SC24-6321
- [z/VM: Saved Segments Planning and Administration](#), SC24-6322
- [z/VM: Secure Configuration Guide](#), SC24-6323

### Customization and Tuning

- [z/VM: CP Exit Customization](#), SC24-6269
- [z/VM: Performance](#), SC24-6301

### Operation and Use

- [z/VM: CMS Commands and Utilities Reference](#), SC24-6260
- [z/VM: CMS Primer](#), SC24-6265
- [z/VM: CMS User's Guide](#), SC24-6266
- [z/VM: CP Commands and Utilities Reference](#), SC24-6268

- [z/VM: System Operation](#), SC24-6326
- [z/VM: Virtual Machine Operation](#), SC24-6334
- [z/VM: XEDIT Commands and Macros Reference](#), SC24-6337
- [z/VM: XEDIT User's Guide](#), SC24-6338

## Application Programming

- [z/VM: CMS Application Development Guide](#), SC24-6256
- [z/VM: CMS Application Development Guide for Assembler](#), SC24-6257
- [z/VM: CMS Application Multitasking](#), SC24-6258
- [z/VM: CMS Callable Services Reference](#), SC24-6259
- [z/VM: CMS Macros and Functions Reference](#), SC24-6262
- [z/VM: CMS Pipelines User's Guide and Reference](#), SC24-6252
- [z/VM: CP Programming Services](#), SC24-6272
- [z/VM: CPI Communications User's Guide](#), SC24-6273
- [z/VM: ESA/XC Principles of Operation](#), SC24-6285
- [z/VM: Language Environment User's Guide](#), SC24-6293
- [z/VM: OpenExtensions Advanced Application Programming Tools](#), SC24-6295
- [z/VM: OpenExtensions Callable Services Reference](#), SC24-6296
- [z/VM: OpenExtensions Commands Reference](#), SC24-6297
- [z/VM: OpenExtensions POSIX Conformance Document](#), GC24-6298
- [z/VM: OpenExtensions User's Guide](#), SC24-6299
- [z/VM: Program Management Binder for CMS](#), SC24-6304
- [z/VM: Reusable Server Kernel Programmer's Guide and Reference](#), SC24-6313
- [z/VM: REXX/VM Reference](#), SC24-6314
- [z/VM: REXX/VM User's Guide](#), SC24-6315
- [z/VM: Systems Management Application Programming](#), SC24-6327
- [z/VM: z/Architecture Extended Configuration \(z/XC\) Principles of Operation](#), SC27-4940

## Diagnosis

- [z/VM: CMS and REXX/VM Messages and Codes](#), GC24-6255
- [z/VM: CP Messages and Codes](#), GC24-6270
- [z/VM: Diagnosis Guide](#), GC24-6280
- [z/VM: Dump Viewing Facility](#), GC24-6284
- [z/VM: Other Components Messages and Codes](#), GC24-6300
- [z/VM: VM Dump Tool](#), GC24-6335

## z/VM Facilities and Features

---

### Data Facility Storage Management Subsystem for z/VM

- [z/VM: DFSMS/VM Customization](#), SC24-6274
- [z/VM: DFSMS/VM Diagnosis Guide](#), GC24-6275
- [z/VM: DFSMS/VM Messages and Codes](#), GC24-6276
- [z/VM: DFSMS/VM Planning Guide](#), SC24-6277



- *z/VM: DFSMS/VM Removable Media Services*, SC24-6278
- *z/VM: DFSMS/VM Storage Administration*, SC24-6279

## Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6281
- *z/VM: Directory Maintenance Facility Messages*, GC24-6282
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283

## Open Systems Adapter

- *Open Systems Adapter/Support Facility on the Hardware Management Console* ([https://www.ibm.com/docs/en/SSLTBW\\_2.3.0/pdf/SC14-7580-02.pdf](https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC14-7580-02.pdf)), SC14-7580
- *Open Systems Adapter-Express ICC 3215 Support* (<https://www.ibm.com/docs/en/zos/2.3.0?topic=osa-icc-3215-support>), SA23-2247
- *Open Systems Adapter Integrated Console Controller User's Guide* ([https://www.ibm.com/docs/en/SSLTBW\\_2.3.0/pdf/SC27-9003-02.pdf](https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC27-9003-02.pdf)), SC27-9003
- *Open Systems Adapter-Express Customer's Guide and Reference* ([https://www.ibm.com/docs/en/SSLTBW\\_2.3.0/pdf/iaa2z1f0.pdf](https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/iaa2z1f0.pdf)), SA22-7935

## Performance Toolkit for z/VM

- *z/VM: Performance Toolkit Guide*, SC24-6302
- *z/VM: Performance Toolkit Reference*, SC24-6303

The following publications contain sections that provide information about z/VM Performance Data Pump, which is licensed with Performance Toolkit for z/VM.

- *z/VM: Performance*, SC24-6301. See *z/VM Performance Data Pump*.
- *z/VM: Other Components Messages and Codes*, GC24-6300. See *Data Pump Messages*.

## RACF Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6305
- *z/VM: RACF Security Server Command Language Reference*, SC24-6306
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6307
- *z/VM: RACF Security Server General User's Guide*, SC24-6308
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6309
- *z/VM: RACF Security Server Messages and Codes*, GC24-6310
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6311
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6312
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6324

## Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6316
- *z/VM: RSCS Networking Exit Customization*, SC24-6317
- *z/VM: RSCS Networking Messages and Codes*, GC24-6318
- *z/VM: RSCS Networking Operation and Use*, SC24-6319
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6320

## TCP/IP for z/VM

- *z/VM: TCP/IP Diagnosis Guide*, GC24-6328
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6329
- *z/VM: TCP/IP Messages and Codes*, GC24-6330
- *z/VM: TCP/IP Planning and Customization*, SC24-6331
- *z/VM: TCP/IP Programmer's Reference*, SC24-6332
- *z/VM: TCP/IP User's Guide*, SC24-6333

## Prerequisite Products

---

### Device Support Facilities

- Device Support Facilities (ICKDSF): User's Guide and Reference ([https://www.ibm.com/docs/en/SSLTBW\\_2.5.0/pdf/ickug00\\_v2r5.pdf](https://www.ibm.com/docs/en/SSLTBW_2.5.0/pdf/ickug00_v2r5.pdf)), GC35-0033

### Environmental Record Editing and Printing Program

- Environmental Record Editing and Printing Program (EREP): Reference ([https://www.ibm.com/docs/en/SSLTBW\\_2.5.0/pdf/ifc2000\\_v2r5.pdf](https://www.ibm.com/docs/en/SSLTBW_2.5.0/pdf/ifc2000_v2r5.pdf)), GC35-0152
- Environmental Record Editing and Printing Program (EREP): User's Guide ([https://www.ibm.com/docs/en/SSLTBW\\_2.5.0/pdf/ifc1000\\_v2r5.pdf](https://www.ibm.com/docs/en/SSLTBW_2.5.0/pdf/ifc1000_v2r5.pdf)), GC35-0151

## Related Products

---

### XL C++ for z/VM

- *XL C/C++ for z/VM: Runtime Library Reference*, SC09-7624
- *XL C/C++ for z/VM: User's Guide*, SC09-7625

### z/OS

IBM Documentation - z/OS (<https://www.ibm.com/docs/en/zos>)

## Other Related Publications

---

This section lists other publications, outside the z/VM library, that you may find helpful.

- *IBM Z: Input/Output Configuration Program User's Guide for ICP IOCP*, SB10-7172.
- *IBM Z: Small Computer Systems Interface (SCSI) IPL Machine Loader Messages*, SC28-7006.

---

# Index

## Special Characters

@ (in national character set) [2](#)  
\*IMG operand  
    LOCATE SPFBK command [627](#)  
\*NLS operand  
    LOCATE SPFBK command [627](#)  
\*NSS operand  
    LOCATE SPFBK command [627](#)  
\*SDF operand  
    LOCATE SPFBK command [628](#)  
# (in national character set) [2](#)  
#CP command [13](#)  
+ operand  
    DEFINE TIMEZONE command [225](#)  
¢ logical line delete symbol [2267](#)  
\$ (in national character set) [2](#)

## Numerics

1403 printer  
    defining [210](#)  
2501 card reader  
    defining [210](#)  
2540 card reader or punch  
    defining [210](#)  
2WCGM operand  
    DEFINE (spooling device) command [211](#)  
3088 CTCA  
    defining [159](#)  
3101 operand  
    TERMINAL TYPE command [1945](#)  
3178 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [1645](#)  
3179 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [1645](#)  
3180 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [1645](#)  
3190 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [1645](#)  
3203 printer  
    changing [1648](#)  
    clearing RDEV information [1636](#)  
    defining [210](#), [1648](#)  
3211 printer  
    defining [210](#)  
3215 display  
    defining as console [149](#)  
3215 operand  
    3215 operand (*continued*)  
        SET RDEVICE command [1661](#)  
3215 terminal  
    changing [1661](#)  
    defining [1661](#)  
3262 printer  
    changing [1648](#)  
    clearing RDEV information [1636](#)  
    defining [210](#), [1648](#)  
3270 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [179](#), [1645](#)  
    defining as console [149](#)  
3277 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [1645](#)  
3278 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [1645](#)  
3279 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [1645](#)  
3290 display  
    changing [1645](#)  
    clearing RDEV information [1636](#)  
    defining [1645](#)  
3370 DASD  
    changing [1641](#)  
    clearing RDEV information [1636](#)  
    defining [1641](#)  
3375 DASD  
    changing [1641](#)  
    clearing RDEV information [1636](#)  
    defining [1641](#)  
3380 DASD  
    changing [1641](#)  
    clearing RDEV information [1636](#)  
    defining [1641](#)  
3390 DASD  
    changing [1641](#)  
    clearing RDEV information [1636](#)  
    defining [1641](#)  
3420 tape unit  
    changing [1658](#)  
    clearing RDEV information [1636](#)  
    defining [1658](#)  
3800 printer  
    changing [1629](#), [1667](#)  
    clearing RDEV information [1636](#)  
    defining [1629](#), [1667](#)  
    using copy modification module [72](#)  
3812 printer  
    changing [1629](#)

3812 printer (*continued*)  
 clearing RDEV information [1636](#)  
 defining [1629](#)

3816 printer  
 changing [1629](#)  
 clearing RDEV information [1636](#)  
 defining [1629](#)

3820 printer  
 changing [1629](#)  
 clearing RDEV information [1636](#)  
 defining [1629](#)

3827 printer  
 changing [1629](#)  
 clearing RDEV information [1636](#)  
 defining [1629](#)

3835 printer  
 changing [1629](#)  
 clearing RDEV information [1636](#)  
 defining [1629](#)

4245 printer  
 changing [1648](#)  
 clearing RDEV information [1636](#)  
 defining [1648](#)

4248 printer  
 changing [1648](#)  
 clearing RDEV information [1636](#)  
 defining [1648](#)

4WCGM operand  
 DEFINE (spooling device) command [211](#)

6262 printer  
 changing [1648](#)  
 clearing RDEV information [1636](#)  
 defining [1648](#)

9332 DASD  
 changing [1641](#)  
 clearing RDEV information [1636](#)  
 defining [1641](#)

9335 DASD  
 changing [1641](#)  
 clearing RDEV information [1636](#)  
 defining [1641](#)

9336 DASD  
 changing [1641](#)  
 clearing RDEV information [1636](#)  
 defining [1641](#)

9345 DASD  
 changing [1641](#)  
 clearing RDEV information [1636](#)  
 defining [1641](#)

## A

abnormal end (abend)  
 dump location, assigning system [1493](#)

access  
 disks  
 by CP [89](#)  
 host system [662](#)  
 modes  
 searching for system logmsg [988](#)  
 system resources, assigning user [1625](#)

access list, host  
 displaying entries [335](#)  
 dumping entries [399](#)

account number  
 changing session [1416](#)

ACCOUNT operand  
 QUERY LOGMSG command [988](#)  
 RECORDING command [1375](#)  
 RETRIEVE utility [2219](#)

accounting record  
 creating [15](#)  
 resetting [15](#)

accounting records, recording  
 changing [1375](#)  
 collecting [2219](#)  
 displaying status [1130](#)

ACI (Access Control Interface)  
 system log messages [988](#)

ACIGROUP operand  
 QUERY LOGMSG command [988](#)

ACNT command [15](#)

activate  
 cache  
 fast write [1427](#)  
 function [1422](#)  
 DASD fast write [1485](#)  
 internal CP checking [1439](#)  
 protected application environment [1434](#)  
 pseudo-page-fault facility [1594](#)  
 trace for real machine events [1446](#)  
 TRACE sets [1980](#)  
 traceid trace [2025](#)

ACTIVATE ISLINK command [16](#)

active  
 IOCDS file  
 changing [1525](#)  
 partition  
 displaying [991](#)

ACTIVE operand  
 QUERY ALL command [759](#)  
 QUERY CTCA command [857](#)  
 QUERY DASD command [868](#)  
 QUERY FCP command [911](#)  
 QUERY GRAF command [931](#)  
 QUERY LINES command [976](#)  
 QUERY OSA command [1055](#)  
 QUERY SWITCHES command [1194](#)  
 QUERY TAPES command [1207](#)  
 QUERY TRACE command [1226](#)  
 QUERY UR command [1252](#)

adapter  
 channel-to-channel  
 changing [1655](#)  
 clearing RDEV information [1636](#)  
 defining [1655](#)

integrated communications  
 changing [1653](#)  
 clearing RDEV information [1636](#)  
 defining [1653](#)

LAN (Local Area Network)  
 changing [1655](#)  
 clearing RDEV information [1636](#)  
 defining [1655](#)

line  
 changing [1638](#)  
 clearing RDEV information [1636](#)  
 defining [1638](#)

- adapter (*continued*)
  - teleprocessing integrated
    - changing [1653](#)
    - clearing RDEV information [1636](#)
    - defining [1653](#)
- add
  - channel path [136](#)
  - channel path to control unit [688](#)
  - communication line [186](#)
  - control unit [162](#)
  - control unit to real device [693](#)
  - DASD volumes
    - to CP-owned volume list [150](#)
  - device [168](#)
  - devices, spool [209](#)
  - display [179](#)
  - information to logo configuration file [1380](#)
  - line, communication [186](#)
  - new
    - alias for existing CP command [122](#), [127](#)
    - CP command [122](#), [143](#)
    - DIAGNOSE code [122](#), [172](#)
    - version of existing CP command [122](#), [143](#)
  - partition to channel path [679](#)
  - partition to real device [693](#)
  - path, channel [136](#)
  - processors to your virtual machine [152](#)
  - real device [168](#)
  - real devices
    - to I/O throttling list [1734](#)
    - to system's definition [1627](#)
  - spool devices [209](#)
  - temporary disk [223](#)
  - unit address to control unit [688](#)
  - virtual
    - communication line [186](#)
    - device [227](#)
    - display [179](#)
    - system console [149](#)
- address
  - displaying for
    - CP command table entry block [603](#), [817](#)
    - data in host storage [630](#)
    - external symbol [636](#)
    - frame table entry [612](#)
    - shared system name table [623](#)
    - spool file block [627](#)
    - system logical device block [616](#)
    - system real device block [618](#)
    - system SNA control block [625](#)
    - user's virtual machine definition block [642](#)
    - VTAM service machine block [647](#)
- ADJUNCT command [18](#), [756](#)
- AFP (advanced function printer)
  - changing [1629](#)
  - clearing RDEV information [1636](#)
  - defining [1629](#)
- AFP operand
  - SET RDEVICE command
    - 3800 printers [1667](#)
    - advanced function printers [1629](#)
    - impact printers [1649](#)
  - START UR operand [1870](#)
- AGENT operand (*continued*)
  - QUERY FCP command [911](#)
  - QUERY OSA command [1056](#)
- alias
  - creating
    - new for existing CP command [122](#), [127](#)
- ALL operand
  - ACNT command [15](#)
  - CLOSE command [71](#)
  - CPU command [107](#)
  - DEFINE (virtual disk) command [230](#)
  - DISABLE (device) command [311](#)
  - DISCARD PINNED command [327](#)
  - DISPLAY PSW command [373](#), [376](#)
  - DISPLAY SCHIB command [383](#)
  - DRAIN (disk) command [386](#)
  - DRAIN UR command [388](#)
  - PURGE IMG command [743](#)
  - PURGE NLS command [745](#)
  - QUERY ALL command [759](#)
  - QUERY ALLOC command [773](#)
  - QUERY CFLINKS command [788](#)
  - QUERY COLLECT command [800](#)
  - QUERY CONV command [810](#)
  - QUERY CTCA command [857](#)
  - QUERY DASD command [868](#)
  - QUERY ENCRYPT command [904](#)
  - QUERY FCP command [911](#)
  - QUERY GATEWAY command [929](#)
  - QUERY GRAF command [931](#)
  - QUERY IMG command [945](#)
  - QUERY ISLINK command [959](#)
  - QUERY LINES command [976](#)
  - QUERY LKFAC command [981](#)
  - QUERY MONITOR command [1014](#)
  - QUERY NLS command [1038](#)
  - QUERY NSS command [1041](#)
  - QUERY OBSERVER command [1052](#)
  - QUERY OSA command [1055](#)
  - QUERY PAV command [1068](#)
  - QUERY RESOURCE command [1138](#)
  - QUERY SECUSER command [1153](#)
  - QUERY SRM command [1173](#)
  - QUERY STGEXEMPT command [1183](#)
  - QUERY SWITCHES command [1194](#)
  - QUERY SYSTEM command [1203](#)
  - QUERY TAPES command [1207](#)
  - QUERY TRACE command [1226](#)
  - QUERY UCR command [1246](#)
  - QUERY UR command [1252](#)
  - QUERY VIRTUAL ALL command [1273](#)
  - RECORDING command [1375](#)
  - SET HOTIO command [1516](#)
  - START (disk) command [1864](#)
  - STOP command [1878](#)
- allocate
  - DASD volume [2124](#)
  - PF key RETRIEVE buffers [1681](#)
  - space for
    - spool records, preventing [386](#)
    - space for internal trace tables [1742](#)
    - virtual processors [107](#)
- allocation
  - cylinder

allocation (*continued*)  
 cylinder (*continued*)  
   displaying [773](#)  
 page  
   displaying [773](#)  
 allow  
   TTY character insert [1927](#)  
 alter  
   3800 printers [1667](#)  
   advanced function printers [1629](#)  
   card  
     punches [1631](#)  
     readers [1634](#)  
   channel path [679](#)  
   channel-to-channel adapter [1655](#)  
   communication controllers [1638](#)  
   control unit [688](#)  
   CP-owned volume list [150](#)  
   CTCA [1655](#)  
   D8ONECMD settings [1482](#)  
   DASD [1641](#)  
   defaults  
     for Stand-Alone Program Loader  
       [2222](#)  
   device [693](#)  
   device numbers, virtual [227](#), [1378](#)  
   emulated devices [1504](#)  
   event data collection profile [711](#)  
   exit [700](#)  
   external attributes of closed spool files [60](#)  
   FCP device [1655](#)  
   graphic display devices [1645](#)  
   HiperSockets device [1655](#)  
   hot I/O rate [938](#), [1516](#)  
   I/O configuration [122](#)  
   impact printers [1648](#)  
   integrated communication adapters [1653](#)  
   IOCDs file for next POR [1525](#)  
   IODC characteristics of real device [693](#)  
   line adapters [1638](#)  
   number of  
     frames for internal trace tables [1742](#)  
     PF key RETRIEVE buffers [1681](#)  
   OSA device [1655](#)  
   path, channel [679](#)  
   privilege class [1616](#)  
   processing options for real DASD [1864](#)  
   PSW data [1901](#)  
   PSWA data [1903](#)  
   queued spool file order [735](#)  
   real device [693](#)  
   real storage [1879](#)  
   record processing parameters [1375](#)  
   sample data collection profile [719](#)  
   secondary user [1688](#)  
   session account number [1416](#)  
   special devices [1655](#)  
   spool options [1808](#)  
   status indicator [734](#)  
   storage size [213](#)  
   system parameters [1705](#)  
   system-resource-access-priority for users [1691](#)  
   system's definition of a set of real devices [1627](#)  
   tape units [1658](#)

alter (*continued*)  
 teleprocessing integrated adapters [1653](#)  
 time zone [225](#), [1738](#)  
 trace traps [1973](#), [2002](#)  
 unsolicited interrupt rate [938](#), [1516](#)  
 unsupported devices [1663](#)  
 user  
   HOLD status of active spool files [60](#)  
   ID of primary system operator [1731](#)  
 virtual  
   configuration [122](#)  
   device numbers [1378](#)  
   machine language [1441](#)  
   processor address [152](#)  
   processor configuration type [1555](#)  
   processor ID [1474](#)  
   virtual machine [1591](#)  
   virtual storage [1879](#)  
 ALTERNATE operand  
   SET TRACEFRAMES command [1742](#)  
 annotate  
   console sheet [12](#)  
   display screen [12](#)  
 ANY option  
   CLOSE command [71](#)  
 ANY3800 operand  
   START UR command [1870](#)  
 AP crypto resources [853](#)  
 APARs  
   displaying [839](#)  
 apvirtual polling [853](#)  
 arrange  
   queued spool file order [735](#)  
 AS operand  
   DEFINE (spooling device) command [210](#)  
   DEFINE (temporary disk) command [223](#)  
   DEFINE (virtual device) command [227](#)  
   DEFINE CONSOLE command [149](#)  
   DEFINE CPU command [152](#)  
   DEFINE CTCA command [159](#)  
   DEFINE GRAF command [179](#)  
   DEFINE LINE command [186](#)  
   DEFINE MDISK command [188](#)  
 ASCII translation [2269](#)  
 ASIT operand  
   LOCK command [654](#)  
   UNLOCK command [2043](#)  
 assign  
   exit point with entry points [20](#)  
   exit point with external symbols [20](#)  
   PF keys [1602](#)  
   RETRIEVE to a PF key [1607](#)  
   system abend dump location [1493](#)  
   user access to system resources [1625](#)  
 associate  
   descriptions with spool files [1914](#)  
   exit point with entry points [20](#)  
   exit point with external symbols [20](#)  
 ASSOCIATE EXIT command [20](#)  
 ASSOCIATE MESSAGES command [24](#)  
 ASSOCIATE MSGS command [24](#)  
 ASTERISK (\*) command [12](#)  
 ASYNCHRONOUS operand  
   CLOSE command [73](#)

ASYNCHRONOUS operand (*continued*)

- CPACCESS command [90](#)
- CPRELEASE command [103](#)
- AT command [27](#)
- attach
  - DASD to host system [32](#)
  - logical devices [32](#)
  - real devices [32](#)
- ATTACH command
  - CRYPTO [32](#)
  - stopping [96](#)
  - SYSASCII [32](#)
- ATTACH operand
  - QUERY ALL command [759](#)
  - QUERY CTCA command [857](#)
  - QUERY DASD command [868](#)
  - QUERY FCP command [912](#)
  - QUERY GRAF command [931](#)
  - QUERY LINES command [976](#)
  - QUERY OSA command [1055](#)
  - QUERY SWITCHES command [1194](#)
  - QUERY TAPES command [1207](#)
  - QUERY UR command [1252](#)
- ATTN command [49](#)
- AUTO operand
  - START UR command [1869](#)
- AUTOLOG command [50](#)
- available
  - nonvolatile storage [1588](#)
  - to you, making other user's devices [578](#)

**B**

- BACKSPACE command [54](#)
- BACKWARD command [54](#)
- BEG3800 operand
  - START UR command [1870](#)
- BEGIN command [56](#)
- BLK operand
  - DEFINE (temporary disk) command [223](#)
  - DEFINE (virtual disk) command [230](#)
- BLKS operand
  - SET VDISK [1749](#)
- block
  - real device, locating [618](#)
  - shared system name [623](#)
  - spool file, locating [627](#)
  - system
    - logical device, locating [616](#)
    - SNA control, locating [625](#)
    - VTAM service machine, locating [647](#)
  - virtual
    - device, locating [639](#)
    - machine definition, locating [642](#)
- block, command table entry
  - defining new CP [143](#)
  - displaying address for CP command [603](#), [817](#)
  - locating for CP command [603](#), [817](#)
- BLOCKS operand
  - SET VDISK [1749](#)
- BOXED operand
  - QUERY ALL command [759](#)
  - QUERY CTCA command [857](#)
  - QUERY DASD command [868](#)

BOXED operand (*continued*)

- QUERY FCP command [912](#)
- QUERY GRAF command [932](#)
- QUERY LINES command [976](#)
- QUERY OSA command [1055](#)
- QUERY SWITCHES command [1194](#)
- QUERY TAPES command [1207](#)
- QUERY UR command [1252](#)
- BRIEF operand
  - QUERY CONV command [811](#)
- BSC\_ADAPTER operand
  - SET RDEVICE command [1638](#)
- BTS operand
  - DEFINE (spooling device) command [211](#)
- buffer, retrieve
  - PF key
    - displaying [1143](#)
    - setting [1681](#)
- BUFFERS operand
  - QUERY RETRIEVE command [1143](#)
- build
  - configuration file of stand-alone dump utility [2235](#)

**C**

- cache
  - fast write data, sending [80](#)
  - file on a minidisk [94](#)
- card punch
  - changing [1631](#)
  - clearing RDEV information [1636](#)
  - defining [1631](#)
- card reader
  - changing [1634](#)
  - clearing RDEV information [1636](#)
  - defining [1634](#)
- CASE operand
  - LOCATE (Storage) command [631](#)
  - LOCATEVM command [651](#)
- CCDUMP utility [2114](#)
- CCLOAD utility [2116](#)
- cease
  - active channel program [520](#)
  - all virtual processors (soft) [1878](#)
  - ATTACH commands [96](#)
  - channel program, active [520](#)
  - console sharing [1688](#)
  - event monitoring [727](#)
  - journal security function [1533](#)
  - LOCATE commands [96](#)
  - LOCATEVM commands [96](#)
  - monitoring of sample data collection [719](#)
  - new operations on real DASD [386](#)
  - processing tape files [1844](#)
  - program, active channel [520](#)
  - sample monitoring [727](#)
  - selected real processors (hard) [1878](#)
  - spool
    - activity on virtual spool devices [70](#)
    - file [498](#)
    - operations on real unit record devices [388](#)
  - system functions [1789](#)
  - tape procedures [1834](#)
  - use of files on minidisk accessed by CP [103](#)

cease (*continued*)  
 VARY commands [96](#)  
 virtual machine session [659](#)

CFS operand  
 DEFINE (spooling device) command [211](#)

change  
 3800 printers [1667](#)  
 advanced function printers [1629](#)  
 card  
   punches [1631](#)  
   readers [1634](#)  
 channel path [679](#)  
 channel-to-channel adapter [1655](#)  
 communication controllers [1638](#)  
 control unit [688](#)  
 CP-owned volume list [150](#)  
 CTCA [1655](#)  
 D8ONECMD settings [1482](#)  
 DASD [1641](#)  
 defaults  
   for Stand-Alone Program Loader  
   [2222](#)  
 device [693](#)  
 device numbers, virtual [227](#), [1378](#)  
 emulated devices [1504](#)  
 event data collection profile [711](#)  
 exit [700](#)  
 external attributes of closed spool files [60](#)  
 FCP device [1655](#)  
 graphic display devices [1645](#)  
 HiperSockets device [1655](#)  
 hot I/O rate [938](#), [1516](#)  
 I/O configuration [122](#)  
 impact printers [1648](#)  
 integrated communication adapters [1653](#)  
 IOCDs file for next POR [1525](#)  
 IODC characteristics of real device [693](#)  
 line adapters [1638](#)  
 number of  
   frames for internal trace tables [1742](#)  
   PF key RETRIEVE buffers [1681](#)  
 OSA device [1655](#)  
 path, channel [679](#)  
 privilege class [1616](#)  
 processing options for real DASD [1864](#)  
 PSW data [1901](#)  
 PSWA data [1903](#)  
 queued spool file order [735](#)  
 real device [693](#)  
 real storage [1879](#)  
 record processing parameters [1375](#)  
 sample data collection profile [719](#)  
 secondary user [1688](#)  
 session account number [1416](#)  
 special devices [1655](#)  
 spool options [1808](#)  
 status indicator [734](#)  
 storage size [213](#)  
 system parameters [1705](#)  
 system-resource-access-priority for users [1691](#)  
 system's definition of a set of real devices [1627](#)  
 tape units [1658](#)  
 teleprocessing integrated adapters [1653](#)  
 time zone [225](#), [1738](#)

change (*continued*)  
 trace traps [1973](#), [2002](#)  
 unsolicited interrupt rate [938](#), [1516](#)  
 unsupported devices [1663](#)  
 user  
   HOLD status of active spool files [60](#)  
   ID of primary system operator [1731](#)  
 virtual  
   configuration [122](#)  
   device numbers [1378](#)  
   machine language [1441](#)  
   processor address [152](#)  
   processor configuration type [1555](#)  
   processor ID [1474](#)  
   virtual machine [1591](#)  
   virtual storage [1879](#)  
 CHANGE command [60](#)

channel path  
 adding  
   to control unit [688](#)  
 changing [679](#)  
 creating [136](#)  
 defining [136](#)  
 deleting  
   from control unit [688](#)  
 disabling [2046](#)  
 displaying  
   devices for logical [791](#), [798](#), [1275](#)  
   status of logical [791](#), [798](#), [1275](#)  
   status of physical [796](#)  
   storage available for adding [895](#)  
 enabling [2046](#)

channel subsystem  
 disabling [2067](#)  
 displaying storage available for adding [895](#)  
 enabling [2067](#)  
 TRACE [1954](#)  
 tracing I/O instructions for [1452](#)

CHARACTERISTICS operand  
 QUERY DASD command [868](#)

CHARS operand  
 CLOSE command [70](#)  
 SET RDEVICE command [1649](#), [1667](#)  
 START UR command [1870](#)

check  
 configuration file syntax [2143](#)  
 controlling frequency of interrupt [1571](#)  
 INDICATE LOAD [536](#)  
 system for a warm start [1789](#)

choose  
 virtual processor [107](#)

class  
 IBM  
   defining for new CP command [143](#)  
   displaying for CP command [603](#), [817](#)  
 privilege  
   changing [1616](#)  
   defining for new CP command [143](#)  
   displaying [1090](#)  
   displaying for CP command [603](#), [817](#)  
 spool file  
   assigning [70](#)  
   specifying for impact printers [1649](#)

CLASS operand



CLASS operand (*continued*)  
     CLOSE command [70](#)  
     QUERY IMG command [945](#)  
     QUERY NLS command [1038](#)  
     QUERY NSS command [1041](#)  
     START UR command [1868](#)  
 CLASSES operand  
     SET RDEVICE command  
         3800 printers [1668](#)  
         card punches [1631](#)  
         card readers [1634](#)  
         impact printers [1649](#)  
 classes, user privilege [6](#)  
 clear  
     device  
         from I/O throttling list [1734](#)  
         information from RDEV [1636](#)  
         pending interrupts [1387](#)  
         RDEV information [1636](#)  
         storage from a real console [1911](#)  
 CLOSE command [70](#)  
 CLOSE option  
     ACNT command [15](#)  
 CMD operand  
     CPU command [107](#)  
 CODE operand  
     SET EMSG command [1512](#)  
 code, DIAGNOSE  
     creating new [122](#), [172](#)  
     defining new [122](#), [172](#)  
     displaying authorized [78](#), [802](#)  
     redefining [697](#)  
     X'08'  
         containing multiple CP commands [865](#), [1482](#)  
         issuing QUERY ALLOC commands through [774](#)  
     X'274'  
         receiving time zone interrupt from [1738](#)  
 collect  
     accounting records [2219](#)  
     error records [2219](#)  
     symptom records [2219](#)  
 command  
     creating  
         new CP [122](#), [143](#)  
         new version of existing CP [122](#), [143](#)  
     displaying authorized [78](#), [802](#)  
     displaying pending [1074](#)  
     redefining CP [683](#)  
 command table entry block  
     defining new CP [143](#)  
     displaying address for CP command [603](#), [817](#)  
     locating for CP command [603](#), [817](#)  
 COMMANDS command [78](#)  
 COMMIT command [80](#)  
 communication controller  
     changing [1638](#)  
     clearing RDEV information [1636](#)  
     defining [1638](#)  
 communication line  
     displaying status [976](#)  
 components, moving to SFS directories [2217](#)  
 CONCOPY (Concurrent Copy)  
     session termination [82](#)  
 CONCOPY TERMINATE command [82](#)  
 CONFIG operand  
     RECORDING command [1375](#)  
     RETRIEVE utility [2219](#)  
 config records, recording  
     changing [1375](#)  
     collecting [2219](#)  
     displaying status [1130](#)  
 configuration file  
     checking syntax of [2143](#)  
     logo  
         adding to [1380](#)  
         replacing [1380](#)  
 configure or deconfigure  
     VARY PCIFUNCTION [2059](#)  
 connect  
     terminal, logically [308](#)  
     user to MPLF configuration [1540](#)  
     virtual CTCA [83](#)  
 console  
     adding system [149](#)  
     changing options [1808](#)  
     sharing, terminating [1688](#)  
     spooling  
         ending [70](#)  
 CONSOLE operand  
     CLOSE command [70](#)  
 content  
     dumping register [433](#)  
 control  
     APL translation table usage [1919](#)  
     attention  
         attention interrupts [1921](#)  
         environment [1936](#)  
     autopoll CP detection [1421](#)  
     break-in [1923](#)  
     carriage return function [1922](#)  
     collection of system data [707](#)  
     CP output destination [1440](#)  
     device prompts [1939](#)  
     display  
         console mode [1928](#), [1941](#)  
         line length [1935](#)  
     display of warning messages [1784](#)  
     error message handling [1512](#)  
     free storage limit detection [1719](#), [1721](#)  
     frequency of interrupt checks [1571](#)  
     host system access [522](#)  
     information, displaying addresses [602](#)  
     informational message display [1519](#)  
     interval timer [1737](#)  
     logical line edit functions [1539](#)  
     messages receipt [1576](#)  
     MONITOR record data [1575](#)  
     NOP CCW simulation [1587](#)  
     pageable storage [654](#)  
     reporting of system data [707](#)  
     response destination [1751](#)  
     screen  
         color [1404](#)  
         highlight [1404](#)  
     selection of system data [707](#)  
     special message receipt [1703](#)  
     spool options [1808](#)  
     SVC76 instruction handling [1729](#)

- control (*continued*)
  - system loader [2271](#)
  - system storage for virtual disks [1749](#)
  - terminal scroll [1940](#)
  - text translation table usage [1943](#)
  - time bomb, virtual machine setting [1736](#)
  - virtual
    - console functions [1917](#)
    - machine response to a command [1683](#)
- control block
  - deleting real device [274](#)
  - displaying address for CP command [603](#), [817](#)
  - locating [602](#)
  - RDEV
    - deleting [274](#)
- control unit
  - adding [162](#)
  - adding to real device [693](#)
  - changing [688](#)
  - creating [162](#)
  - defining [162](#)
  - deleting [263](#)
  - deleting from real device [693](#)
  - displaying storage available for adding [895](#)
- controller
  - changing [1638](#)
  - clearing RDEV information [1636](#)
  - defining [1638](#)
- copy
  - display screen [1605](#)
  - user minidisks [2149](#)
- COPY option
  - CLOSE command [70](#)
- core, adding or removing [2048](#)
- count
  - successful trace events [1972](#)
- COUNT operand
  - LOCATE (Storage) command [631](#)
  - LOCATEVM command [652](#)
  - QUERY IMG command [945](#)
  - QUERY NLS command [1038](#)
  - QUERY NSS command [1042](#)
  - QUERY UCR command [1246](#)
- COUNTS operand
  - QUERY RETRIEVE command [1143](#)
- COUPLE command [83](#)
- CP (Control Program)
  - causing to open file on minidisk [94](#)
  - command
    - defining new [143](#)
    - redefining [683](#)
  - disk access
    - displaying [824](#)
    - establishing [89](#)
  - module
    - displaying location of [832](#)
  - nucleus
    - displaying location of [832](#)
  - storage
    - locating data in [630](#)
  - system execution space
    - loading CP routines into [111](#)
    - unloading CP routines from [117](#)
- CP command [88](#)
- CP-owned volume list
  - changing [150](#)
  - defining [150](#)
  - displaying [834](#)
- CPACCESS command [89](#)
- CPCACHE command [94](#)
- CPEREPXA utility [2118](#)
- CPFMTXA utility [2124](#)
- CPHX command [96](#)
- CPLISTFILE command [99](#)
- CPRELEASE command [103](#)
- CPSYNTAX utility [2143](#)
- CPTYPE command [105](#)
- CPU (central processing unit)
  - defining [152](#)
  - designating [107](#)
- CPU command [107](#)
- CPVLoad command [109](#)
- CPVLOAD command [109](#)
- CPXLOAD command [111](#)
- CPXUNLOAD command [117](#)
- create
  - channel path [136](#)
  - control unit [162](#)
  - device [168](#)
  - image library [2201](#)
  - new
    - alias for existing CP command [122](#), [127](#)
    - CP command [122](#), [143](#)
    - DIAGNOSE code [122](#), [172](#)
    - guest LAN [122](#)
    - version of existing CP command [122](#), [143](#)
  - path, channel [136](#)
  - pending attention interrupt [1386](#)
  - real device [168](#)
  - text decks [2192](#)
  - user account data [15](#)
  - user directory [2168](#)
- cross-system linking
  - XLINK CHECK command [2110](#)
  - XLINK command [2108](#)
  - XLINK DISPLAY utility [2257](#)
  - XLINK FINDTAB command [2111](#)
  - XLINK FORMAT utility [2260](#)
  - XLINK RESET command [2112](#)
- CTCA (channel-to-channel adapter)
  - changing [1655](#)
  - clearing RDEV information [1636](#)
  - defining [159](#), [1655](#)
  - displaying status [857](#)
- CTCA operand
  - DEFINE CTCA command [159](#)
  - SET RDEVICE command [1655](#)
- CVTOVRID utility [2147](#)
- CYL operand
  - DEFINE (temporary disk) command [223](#)
- cylinder
  - allocation
    - displaying [773](#)

## D

- D8ONECMD settings
  - changing [1482](#)

D8ONECMD settings (*continued*)  
 displaying [865](#)

DASD (Direct Access Storage Device)  
 adding to CP-owned volume list [150](#)  
 attaching to host system [32](#)  
 changing [1641](#)  
 changing in CP-owned volume list [150](#)  
 changing processing options [1864](#)  
 clearing RDEV information [1636](#)  
 CP-owned  
   displaying list [834](#)  
 defining [1641](#)  
 displaying  
   cylinder or page allocation [773](#)  
   status [867](#)  
 fast write function, setting [1485](#)  
 Parallel Access Volume device  
   displaying status [867](#), [1068](#)  
 preventing new operations [386](#)  
 restarting [1864](#)  
 stopping new operations [386](#)  
 unsupported  
   changing [1663](#)  
   clearing RDEV information [1636](#)  
   defining [1663](#)  
 used by UTILITY utility [2255](#)

DASD operand  
 DRAIN (disk) command [386](#)  
 QUERY CPOWNER command [834](#)  
 SET HOTIO command [1516](#)  
 SET RDEVICE command [1641](#), [1664](#)  
 START (disk) command [1864](#)

data  
 searching for in virtual machine's storage [651](#)  
 storing at specified locations (ESA/XC, z/XC) [1881](#)  
 tracing [2018](#)

data collection  
 profile, changing event [711](#)  
 profile, changing sample [719](#)  
 profile, setting event [711](#)  
 profile, setting sample [719](#)

DATA operand  
 LOCATE (Storage) command [631](#)  
 LOCATEVM command [652](#)

DBCS (Double-Byte Character Set)  
 data supported  
   MESSAGE ALL command [674](#)  
   MSGNOH command [731](#)  
   SEND command [1407](#)  
   SET LOGMSG command [1551](#)  
   SET PFnn command [1602](#)  
   SMSG command [1800](#)  
   TERMINAL APL [1919](#)  
   TERMINAL LINESIZE command [1935](#)  
   TERMINAL TEXT [1943](#)  
   WARNING [2097](#)  
 TERMINAL APL [1919](#)

DDR utility [2149](#)

deactivate  
 cache fast write [1427](#)  
 cache function [1422](#)  
 DASD fast write [1485](#)  
 internal CP checking [1439](#)  
 protected application environment [1434](#)

deactivate (*continued*)  
 pseudo-page-fault facility [1594](#)  
 trace for real machine events [1446](#)

DEACTIVE CONV command [119](#)  
 DEACTIVE ISLINK command [120](#)

DECREMENT operand  
 LOCATE (Storage) command [631](#)  
 LOCATEVM command [652](#)

dedicated advanced function printer  
 changing [1629](#)  
 clearing RDEV information [1636](#)  
 defining [1629](#)

dedicated minidisk  
 displaying device characteristics [1004](#)

DEFAULT operand  
 SET HOTIO command [1517](#)

DEFFCB operand  
 START UR command [1871](#)

define  
 3800 printers [1667](#)  
 advanced function printers [1629](#)  
 break key [1924](#)  
 card  
   punches [1631](#)  
   readers [1634](#)  
 channel path [136](#)  
 channel-to-channel adapter [159](#), [1655](#)  
 communication  
   controllers [1638](#)  
   line [186](#)  
 control unit [162](#)  
 CP-owned volume list [150](#)  
 CTCA [159](#), [1655](#)  
 DASD [1641](#)  
 DASD fast write [1485](#)  
 device [168](#)  
 device recording mode [1671](#)  
 device-end interrupt pending [1374](#)  
 devices  
   spool [209](#)  
   terminals [1945](#)  
 emulated devices [1504](#)  
 error message handling [1512](#)  
 error recording mode [1574](#)  
 event data collection profile [711](#)  
 exit [176](#)  
 FCP device [1655](#)  
 graphic display devices [1645](#)  
 HiperSockets device [1655](#)  
 hot I/O rate [1516](#)  
 impact printers [1648](#)  
 informational message display [1519](#)  
 integrated communication adapters [1653](#)  
 internal CP checking [1439](#)  
 interrupt rate, unsolicited [1516](#)  
 IOCDs file for next POR [1525](#)  
 line  
   adapters [1638](#)  
   communication [186](#)  
 logical  
   character-delete symbol [1926](#)  
   escape symbol [1930](#)  
   line-delete symbol [1933](#)  
   line-end symbol [1934](#)

define (*continued*)

logical (*continued*)

tab characters [1942](#)

logical line edit functions [1539](#)

maximum number of logged on users [1559](#)

message processor [191](#)

new

alias for existing CP command [122](#), [127](#)

CP command [122](#), [143](#)

DIAGNOSE code [122](#), [172](#)

version of existing CP command [122](#), [143](#)

number of PF key RETRIEVE buffers [1681](#)

OSA device [1655](#)

password command entry [1598](#)

path, channel [136](#)

PAVALIAS [198](#)

products, using SET PRODUCT [1619](#)

protected application environment [1434](#)

real

PCI function [199](#)

real device [168](#)

real devices [1627](#)

relocation domain [1755](#)

reserved pages (frames) [1674](#)

sample data collection profile [719](#)

secondary user [1688](#)

SET OBSERVER command [1591](#)

shutdown signal disablement [1700](#)

skeleton SDF for

NSS [253](#)

saved segment [246](#)

special devices [1655](#)

spool

devices [209](#)

spool file class for impact printers [1649](#)

storage size [213](#)

system abend dump location [1493](#)

system storage for virtual disks [1749](#)

tab function values [1609](#)

tape units [1658](#)

teleprocessing integrated adapters [1653](#)

temporary disk [223](#)

time zone [225](#)

trace

guest [2018](#)

unsolicited interrupt rate [1516](#)

unsupported devices [1663](#)

user access to system resources [1625](#)

user frame owned list reordering [1672](#)

user ID of primary system operator [1731](#)

virtual

channel-to-channel adapter [159](#)

communication line [186](#)

CTCA [159](#)

device [227](#)

disks [230](#)

switches [232](#)

virtual processor ID [1474](#)

DEFINE (spooling device) command [209](#)

DEFINE (temporary disk) command [223](#)

DEFINE (virtual device) command [227](#)

DEFINE (virtual disk) command [230](#)

DEFINE ALIAS command [127](#)

DEFINE CFLINK [130](#)

DEFINE CHPID command [136](#)

DEFINE CNTLUNIT command [162](#)

DEFINE command [122](#)

DEFINE COMMAND command [143](#)

DEFINE CONSOLE command [149](#)

DEFINE CPOWNER command [150](#)

DEFINE CPU command [152](#)

DEFINE CPUPOOL command superseded by DEFINE RESPOOL command [205](#)

DEFINE CRYPTO command [157](#)

DEFINE CTCA command [159](#)

DEFINE CU command [162](#)

DEFINE DEVICE command [168](#)

DEFINE DIAGNOSE command [172](#)

DEFINE EXIT command [176](#)

DEFINE GRAF command [179](#)

DEFINE HYPERPAVALIAS command [180](#)

DEFINE IODEVICE command [168](#)

DEFINE LAN command [181](#)

DEFINE LINE command [186](#)

DEFINE LPAR command [187](#)

DEFINE MDISK command [180](#), [188](#)

DEFINE NIC command [194](#)

DEFINE PATH command [136](#)

DEFINE PAVALIAS command [198](#)

DEFINE PCIFUNCTION command [199](#)

DEFINE RELODOMAIN command [203](#)

DEFINE RESPOOL command [205](#)

DEFINE STORAGE command [213](#)

DEFINE SUBSTITUTE command [222](#)

DEFINE TIMEZONE command [225](#)

DEFINE VSWITCH command [232](#)

defining virtual switch [232](#)

definition of

set of real devices, adding to [1627](#)

set of real devices, changing [1627](#)

DEFSEG command [246](#)

DEFSYS command [253](#)

delayed purge queue

specifying maximum size [1668](#)

delete

all named trace set traps [1971](#)

all trace traps and sets [1974](#)

channel path [259](#)

channel path from control unit [688](#)

closed spool files [740](#)

control unit [263](#)

control unit from real device [693](#)

devices [265](#)

emulated devices [268](#), [898](#)

entire log message [1551](#)

guest operating system messages [2069](#)

image library files [743](#)

log message by line [1551](#)

message repository files [745](#)

NSS files [747](#)

partition from channel path [679](#)

partition from real device [693](#)

path, channel [259](#)

RDEV [274](#)

real

PCI function [272](#)

real device

control block [274](#)

delete (*continued*)

- real device (*continued*)
  - from I/O configuration [265](#)
  - from I/O throttling list [1734](#)
- real device control block [274](#)
- system data files [747](#)
- system trace files [749](#)
- trace traps [2002](#)
- traceids [2024](#)
- UCR files [751](#)
- unit address from control unit [688](#)
- unit, control [263](#)

DELETE CHPID command [259](#)

DELETE CNTLUNIT command [263](#)

DELETE CPUPOOL command superseded by DELETE RESPOOL command [276](#)

DELETE CU command [263](#)

DELETE DEVICE command [265](#)

DELETE EDEVICE command [268](#)

DELETE IODEVICE command [265](#)

DELETE LPAR command [270](#)

DELETE PATH command [259](#)

DELETE PCIFUNCTION command [272](#)

DELETE RDEVICE command [274](#)

DELETE RESPOOL command [276](#)

DEST operand

- CLOSE command [70](#)
- SET RDEVICE command [1650](#)
- START UR command [1868](#)

DESTAGE command [277](#)

destination

- controlling CP output [1440](#)

detach

- local or remote devices [296](#)
- message processor [292](#)
- processors using DETACH CPU [284](#)
- virtual devices [304](#)

DETACH (real device) command [296](#)

DETACH (virtual device) command [304](#)

DETACH CFLINK [282](#)

DETACH command [279](#)

DETACH CPU command [284](#)

DETACH CRYPTO command [286](#)

DETACH LAN command [290](#)

DETACH MSGPROC [292](#)

DETACH NIC command [293](#)

detach pcifunction

- virtual machine [294](#)

DETACH PCIFUNCTION command [294](#)

DETACH SYSASCSII command [302](#)

detaching virtual switch segment [307](#)

DETAIL operand

- QUERY CONV command [811](#)

DETAILS operand

- QUERY DASD command [868](#)
- QUERY MDISK command [1004](#)
- QUERY TAPES command [1207](#)

determine

- CHPID virtualization directory setting [798](#), [1275](#)
- devices on channel path [791](#)
- logical status of channel path [791](#)
- MONITOR record data [1575](#)
- physical status of channel path [796](#)

determine (*continued*)

- status of channel path [791](#), [796](#), [798](#), [1275](#)
- tape file names [1857](#)
- volumes under cross-system link control [2110](#)

DEVCLASS operand

- SET RDEVICE command [1664](#)

DEVCTL operand

- ATTACH command [35](#)

device

- adding emulated [1504](#)
- adding real [168](#), [1627](#)
- block
  - real, locating [618](#)
  - system logical, locating [616](#)
- changing [693](#)
- changing emulated [1504](#)
- changing real [1627](#)
- characteristics of minidisks [1004](#)
- control block, deleting real [274](#)
- creating real [168](#)
- defining real [168](#)
- deleting emulated [268](#)
- deleting real [265](#)
- deleting real control block [274](#)
- displaying
  - emulated [898](#)
  - installed on channel path [791](#), [798](#), [1275](#)
  - storage available for adding [895](#)
  - whether CP initializes new [1033](#)
- initializing
  - after IOMCK [1581](#)
  - when added to the system [1581](#)
- new, displaying whether CP initializes [1033](#)
- path, displaying installed [1064](#)
- restarting drained spool [1867](#)

DEVICE operand

- SET CACHE command [1422](#)
- SET DASDFW command [1485](#)

Device Support Facilities (ICKDSF) [2129](#)

DIAGNOSE code

- creating new [122](#), [172](#)
- defining new [122](#), [172](#)
- displaying authorized [78](#), [802](#)
- redefining [697](#)
- X'08'
  - containing multiple CP commands [865](#), [1482](#)
  - issuing QUERY ALLOC commands through [774](#)
- X'274'
  - receiving time zone interrupt from [1738](#)

DIAL command [308](#)

direct access subsystem cache [277](#)

DIRECTORY operand

- QUERY MDISK command [1004](#)

DIRECTXA utility [2168](#)

disable

- cache fast write [1427](#)
- cache function [1422](#)
- channel path [2046](#)
- DASD fast write [1485](#)
- devices for host system access [311](#)
- exit [319](#)
- HCD [321](#)
- I/O throttling [1734](#)
- internal CP checking [1439](#)

disable (*continued*)

- logical connections to real devices [2053](#)
- path, channel [2046](#)
- products using SET PRODUCT [1619](#)
- products, displaying status of [1095](#)
- protected application environment [1434](#)
- pseudo-page-fault facility [1594](#)
- real device [2063](#)
- subchannel [2067](#)
- subchannel measuring [1685](#)
- trace for real machine events [1446](#)
- trace ID tracing [2022](#)
- DISABLE (device) command [311](#)
- DISABLE CMD command [313](#)
- DISABLE COMMAND command [313](#)
- DISABLE DIAGNOSE command [316](#)
- DISABLE EXITS command [319](#)
- DISABLE HCD command [321](#)
- DISASSOCIATE command [324](#)
- DISCARD PINNED command [327](#)
- disconnect
  - from host system [329](#)
  - terminal logically [2042](#)
  - user to MPLF configuration [1540](#)
- DISCONNECT command [329](#)
- disk
  - CP-owned
    - changing [150](#)
    - defining [150](#)
    - displaying cylinder or page allocation [773](#)
    - displaying list [834](#)
  - preventing new operations [386](#)
  - stopping new operations [386](#)
- DISKMAP utility [2178](#)
- display
  - abend dump destinations [887](#)
  - ABEND SOFT status [753](#)
  - access modes to search for system logmsg [988](#)
  - active partition [991](#)
  - active users [532](#)
  - address
    - control information [602](#)
    - CP command table entry block [603](#), [817](#)
    - address of real PCI function block [621](#)
    - address of virtual PCI function block [645](#)
    - address space information [1170](#)
    - addresses of control information [651](#)
  - all
    - partitions [991](#)
    - paths installed to a device [1064](#)
  - APARs [839](#)
  - associated control block [621](#), [645](#)
  - authorized commands [78](#), [802](#)
  - authorized diagnose codes [78](#)
  - authorized DIAGNOSE codes [802](#)
  - AVS information [929](#)
  - buffers, PF key retrieve [1143](#)
  - BYUSER ID of a specified user [781](#)
  - cache fast write status [784](#)
  - cache status [782](#)
  - capability value of processors [786](#)
  - channel-to-channel adapter status [1287](#)
  - channel-to-channel adapters [857](#)
  - chargeable account [755](#)

display (*continued*)

- command limit threshold [799](#)
- commands, pending [1074](#)
- communication lines [976](#)
- concurrent copy session information [804](#)
- configuration mode setting [806](#)
- console color [1151](#)
- console status [1278](#)
- controlling warning message [1784](#)
- conversations managed by node [810](#)
- CP attributes [829](#), [1488](#)
- CP data collection status [1130](#)
- CP message language [828](#)
- CP-owned volume list [834](#)
- cryptographic facility status [1284](#)
- CS data link information [959](#)
- current session information [1219](#)
- current time zone definitions [1222](#)
- cylinder allocation [773](#)
- D8ONECMD settings [865](#)
- DASD [867](#)
- DASD data tracks [1079](#)
- DASD fast write status [879](#)
- date [1219](#)
- default reorder setting [1134](#)
- device settings information [883](#)
- devices
  - changing [1645](#)
  - clearing RDEV information [1636](#)
  - controlling monitor data from [1575](#)
  - defining [1645](#)
  - displaying status [931](#)
  - on channel path [791](#)
  - unsupported, changing [1663](#)
  - unsupported, clearing RDEV information [1636](#)
  - unsupported, defining [1663](#)
  - whether CP initializes new [1033](#)
- dispatch lists [548](#)
- display devices [931](#)
- domain information [800](#)
- dump options [887](#)
- dynamic switching devices [1194](#)
- eligible lists [548](#)
- environmental characteristics [531](#)
- existing virtual disks in storage [1271](#)
- FCP [911](#), [1297](#)
- first-level storage [363](#)
- free storage limit detection [1183](#), [1185](#)
- global resource information [1138](#)
- guest page 0 [372](#)
- guest storage (ESA/390) [345](#)
- guest storage (ESA/XC, z/XC) [337](#)
- guest storage (z/Architecture) [354](#)
- HCD status [934](#)
- highlight values [1151](#)
- host access list entries [335](#)
- host storage [363](#)
- hot I/O rate [938](#)
- HYPERSWAP status [941](#)
- image library information [945](#)
- information about trace file(s) [1231](#)
- interrupt detector intervals [1011](#)
- interrupt rate, unsolicited [938](#)
- IPL parameters, system [954](#)

display (*continued*)

- key retrieve buffers, PF [1143](#)
- language identifiers [826](#)
- limit of allowed users [999](#)
- line-mode output [1052](#)
- link indicators in map records [2257](#)
- link information [979](#)
- list of internal fences [915](#)
- listed users [535](#)
- local modifications [839](#)
- local resource information [1138](#)
- location
  - control information [602](#)
  - CP command table entry block [603](#), [817](#)
  - CP module [832](#)
  - CP nucleus [832](#)
  - data in host storage [630](#)
  - external symbol address [636](#)
  - parm disk [832](#)
  - shared system name table [623](#)
  - spool file block [627](#)
  - system frame table entry [612](#)
  - system logical device block [616](#)
  - system real device block [618](#)
  - system SNA control block [625](#)
  - user's spool file identification entry [610](#)
  - user's virtual machine definition block [642](#)
  - VTAM service machine block [647](#)
- log message [988](#)
- logged-on user IDs [1029](#)
- message repository file information [1038](#)
- minidisk device characteristics [1004](#)
- monitor profile [1014](#)
- MPLF simulation configuration [981](#)
- NSS files information [1041](#)
- number of logged-on users [1264](#)
- number of reserved host real storage frames [1135](#)
- number of users [535](#)
- number of users dialed [1264](#)
- open spool file [1341](#)
- Open Systems Adapter [1316](#)
- Open Systems Adapters [1055](#)
- operating load [535](#)
- output mode [1341](#)
- page allocation [773](#)
- page residency data [546](#)
- page wait status [546](#)
- paging rate [535](#)
- partitions [991](#)
- password suppression status [1062](#)
- pending commands [1074](#)
- PF key
  - retrieve buffers [1143](#)
- PF key assignments [1077](#)
- prefix register [372](#)
- privilege class [1090](#)
- processor identification [846](#)
- processor utilization [535](#)
- PRODUCT statement information [1095](#)
- PSW information [373](#), [376](#)
- PTFs [839](#)
- QUERY DATEFORMAT [881](#)
- QUICKDSP attribute [1102](#)
- real device status [1114](#)

display (*continued*)

- real machine component data [331](#)
- real processor utilization [1092](#)
- real storage size [1186](#)
- real storage status [925](#)
- register contents [378](#)
- relocation domain [203](#), [1132](#), [1348](#)
- remote service access word [1145](#)
- resource measurement [531](#)
- resources used by virtual machine [555](#)
- retrieve buffers, PF key [1143](#)
- RSAW [1145](#)
- scheduler parameters [1173](#)
- SCHIB information [383](#)
- secondary user [1153](#)
- selected file lines [105](#)
- server time protocol [1191](#)
- SET functions [1156](#)
- setting informational message [1519](#)
- shared segments [543](#)
- space allocated for internal trace tables [1230](#)
- spool file count [917](#)
- spool file information [1104](#)
- spool file limits [997](#)
- state entries of guest linkage stack [368](#)
- status
  - all devices [759](#)
  - channel path [791](#), [796](#), [798](#), [1275](#)
  - CP-accessed disk [824](#)
  - cryptographic facility [853](#), [1284](#)
  - duplex pairs of devices [893](#)
  - dynamic I/O support [895](#)
  - host real storage [759](#)
  - installed device paths [1064](#)
  - journaling, security [963](#)
  - logical devices [973](#)
  - measuring, subchannel [1149](#)
  - Parallel Access Volume DASD [867](#), [1068](#)
  - print server virtual machine [1089](#)
  - processor device [1291](#)
  - real channel-to-channel adapters [857](#)
  - real communication lines [976](#)
  - real DASD [867](#)
  - real devices [1114](#)
  - real display devices [931](#)
  - real dynamic switching devices [1194](#)
  - real magnetic tape drives [1207](#)
  - security journaling [963](#)
  - subchannel measuring [1149](#)
  - switch devices [1336](#)
  - unit record device [1340](#)
  - unit record devices [1252](#)
  - UR [1340](#)
  - virtual communication lines [1305](#)
  - virtual DASD [1288](#)
  - virtual device number [1291](#)
  - virtual displays [1303](#)
  - virtual duplex pairs [1295](#)
  - virtual printer [1324](#)
  - virtual punch [1328](#)
  - virtual reader [1331](#)
  - virtual tape devices [1338](#)
- status data [1239](#)
- status of internal CP checking [816](#)

display (*continued*)

- status of nonvolatile storage [1050](#)
- storage
  - available for dynamic I/O changes [895](#)
- storage status, real [925](#)
- subchannel information blocks [383](#)
- system data file count [1152](#)
- system data file information [1041](#)
- system execution space page status [1196](#)
- system execution space, address ranges used for [1199](#)
- system gateway information [929](#)
- system information [929](#)
- system operator [1202](#)
- system resource contention [531](#)
- system resource information [1138](#)
- system resources [531](#)
- system scheduler/dispatcher [531](#)
- tag information [1205](#)
- tape file names [1857](#)
- TDISK clearing [1212](#)
- temporary disk space [1211](#)
- terminal event record information [1013](#)
- terminal options [1213](#)
- the count of trace files (all users) [1231](#)
- the count of trace files (one user) [1231](#)
- time [1219](#)
- trace destination [1235](#)
- trace function status [842](#)
- trace settings [1226](#)
- TRACE settings [1076](#)
- trace status [1235](#)
- traceback table [1999](#)
- traceid status [2023](#)
- UCR file information [1246](#)
- unsolicited interrupt rate [938](#)
- user access to system DASD [1203](#)
- user device number [1264](#)
- user ID and system identifier [1261](#)
- user ID of system operator [1202](#)
- user IDs of virtual machines [531](#)
- users in dispatch, eligible and dormant lists [532](#)
- users in logon hold status [936](#)
- users with output in system hold status [936](#)
- valid CP languages [826](#)
- virtual CPUS [1282](#)
- virtual machine component data [331](#)
- virtual machine resources [555](#)
- virtual machine spaces [552](#)
- virtual machine status [1273](#)
- virtual machines in I/O wait state [534](#)
- virtual storage (ESA/390) [345](#)
- virtual storage (ESA/XC, z/XC) [337](#)
- virtual storage (z/Architecture) [354](#)
- virtual storage size [1333](#)
- whether CP initializes new devices [1033](#)
- your share of the system [1161](#)
- DISPLAY (Guest Storage - ESA/390) command [345](#)
- DISPLAY (Guest Storage - ESA/XC, z/XC) command [337](#)
- DISPLAY (Guest Storage - z/Architecture) command [354](#)
- DISPLAY (Host Storage) command [363](#)
- DISPLAY ACCESS LIST command [335](#)
- DISPLAY command [331](#)
- DISPLAY Linkage Stack command [368](#)
- DISPLAY operand
  - SET RDEVICE command [1664](#)
- DISPLAY PREFIX command [372](#)
- DISPLAY PSW command [373](#), [376](#)
- DISPLAY Registers command [378](#)
- DISPLAY SCHIB command [383](#)
- DIST option
  - CLOSE command [71](#)
- domains [853](#)
- DPMSIZE operand
  - SET RDEVICE command [1668](#)
- DPS operand
  - SET RDEVICE command [1664](#)
- DRAIN (disk) command [386](#)
- DRAIN UR command [388](#)
- drained spool device, restarting [1867](#)
- DRAINING operand
  - QUERY DASD command [869](#)
- DRCT operand
  - QUERY ALLOC command [773](#)
  - QUERY MDISK command [1004](#)
- drive, magnetic tape, displaying status [1207](#)
- DUMP (Guest Storage - ESA/390) command [408](#)
- DUMP (Guest Storage - ESA/XC, z/XC) command [401](#)
- DUMP (Guest Storage - z/Architecture) command [416](#)
- DUMP (Host Storage) command [424](#)
- DUMP ACCESS LIST command [399](#)
- DUMP command [395](#)
- DUMP Linkage Stack command [429](#)
- DUMP operand
  - DEFINE CPOWNER command [150](#)
- DUMP PREFIX command [433](#)
- DUMP PSW command [434](#), [437](#)
- DUMP Registers command [439](#)
- DUMP SCHIB command [445](#)
- dumping
  - communication controller [2116](#)
  - entries in host access list [399](#)
  - files to tape [1836](#)
  - first-level storage [424](#)
  - guest linkage stack state entries [429](#)
  - guest storage (ESA/390) [408](#)
  - guest storage (ESA/XC, z/XC) [401](#)
  - guest storage (z/Architecture) [416](#)
  - host storage [424](#)
  - prefix register contents [433](#)
  - PSW information [434](#), [437](#)
  - real machine data [395](#)
  - register contents [439](#)
  - SCHIB information [445](#)
  - user minidisks [2149](#)
  - virtual machine data [395](#)
  - virtual storage (ESA/390) [408](#)
  - virtual storage (ESA/XC, z/XC) [401](#)
  - virtual storage (z/Architecture) [416](#)
- dumping second-level storage
  - using VMDUMP [2073](#)
- DUMPLD2 utility [2184](#)
- DUMpload utility [2180](#)
- DUPLEX command [448](#)
- dynamic I/O
  - commands
    - DEFINE CHPID [136](#)



dynamic I/O (*continued*)  
 commands (*continued*)  
 DEFINE CNTLUNIT [162](#)  
 DEFINE CU [162](#)  
 DEFINE DEVICE [168](#)  
 DEFINE IODEVICE [168](#)  
 DEFINE PATH [136](#)  
 DELETE CHPID [259](#)  
 DELETE CNTLUNIT [263](#)  
 DELETE CU [263](#)  
 DELETE DEVICE [265](#)  
 DELETE IODEVICE [265](#)  
 DELETE PATH [259](#)  
 MODIFY CHPID [679](#)  
 MODIFY CNTLUNIT [688](#)  
 MODIFY CU [688](#)  
 MODIFY DEVICE [693](#)  
 MODIFY IODEVICE [693](#)  
 MODIFY PATH [679](#)  
 QUERY CONFIGMODE [806](#)  
 QUERY DYNAMIC\_I/O [895](#)  
 SET CONFIGMODE [1436](#)  
 SET DEVICES [1490](#)  
 SET DYNAMIC\_I/O [1502](#)  
 using in configuration mode [1436](#)  
 displaying  
 status [895](#)  
 storage [895](#)  
 dynamic switching device  
 displaying status [1194](#)  
 unsupported  
 changing [1663](#)  
 clearing RDEV information [1636](#)  
 defining [1663](#)

## E

EAST operand  
 DEFINE TIMEZONE command [225](#)  
 ECHO command [457](#)  
 EMULATED\_3270 operand  
 SET RDEVICE command [1645](#)  
 enable  
 cache fast write [1427](#)  
 cache function [1422](#)  
 channel path [2046](#)  
 DASD fast write [1485](#)  
 devices for host system access [458](#)  
 exit [465](#)  
 I/O throttling [1734](#)  
 internal CP checking [1439](#)  
 logical connections to real devices [2053](#)  
 path, channel [2046](#)  
 products using SET PRODUCT [1619](#)  
 products, displaying status of [1095](#)  
 protected application environment [1434](#)  
 pseudo-page-fault facility [1594](#)  
 subchannel [2067](#)  
 subchannel measuring [1685](#)  
 trace for real machine events [1446](#)  
 ENABLE CMD command [460](#)  
 ENABLE command [458](#)  
 ENABLE COMMAND command [460](#)  
 ENABLE DIAGNOSE command [463](#)

ENABLE EXITS command [465](#)  
 enabling a device  
 using VARY [2063](#)  
 ENCRYPTION\_STATUS operand  
 QUERY FCP command [912](#)  
 end  
 active channel program [520](#)  
 all virtual processors (soft) [1878](#)  
 ATTACH commands [96](#)  
 channel program, active [520](#)  
 console sharing [1688](#)  
 event monitoring [727](#)  
 journal security function [1533](#)  
 LOCATE commands [96](#)  
 LOCATEVM commands [96](#)  
 monitoring of sample data collection [719](#)  
 new operations on real DASD [386](#)  
 processing tape files [1844](#)  
 program, active channel [520](#)  
 sample monitoring [727](#)  
 selected real processors (hard) [1878](#)  
 spool  
 activity on virtual spool devices [70](#)  
 file [498](#)  
 operations on real unit record devices [388](#)  
 system functions [1789](#)  
 tape procedures [1834](#)  
 use of files on minidisk accessed by CP [103](#)  
 VARY commands [96](#)  
 virtual machine session [659](#)  
 END operand  
 DEFINE MDISK command [188](#)  
 LOCATE (Storage) command [630](#)  
 LOCATEVM command [651](#)  
 enter  
 echo environment [457](#)  
 entry  
 frame table, locating [612](#)  
 spool file identification, locating [610](#)  
 entry block, command table  
 defining new CP [143](#)  
 displaying address for CP command [603](#), [817](#)  
 locating for CP command [603](#), [817](#)  
 ENTRY operand  
 LOCATE FRAMETBL command [612](#)  
 environment  
 entering echo [457](#)  
 EOF option  
 CLOSE command [71](#)  
 EREP operand  
 RECORDING command [1375](#)  
 RETRIEVE utility [2219](#)  
 EREP records, recording  
 changing [1375](#)  
 collecting [2219](#)  
 displaying status [1130](#)  
 error records, routing [2219](#)  
 establish  
 filemode [89](#)  
 event  
 data collection profile, changing [711](#)  
 data collection profile, setting [711](#)  
 monitoring [727](#)  
 example

example (continued)

- alias, creating for CP command [129](#)
- assigning entry points to exit points [22](#)
- changing
  - channel paths dynamically [681](#)
  - control units dynamically [691](#)
  - device definitions dynamically, real [695](#)
  - paths dynamically, channel [681](#)
  - real device definitions dynamically [695](#)
- changing IPL record [1531](#)
- changing privilege classes [686](#), [699](#)
- channel path
  - changing dynamically [681](#)
  - defining dynamically [142](#)
  - deleting dynamically [260](#)
  - displaying status [792](#), [796](#), [798](#), [1275](#)
- collecting subchannel measurement data [1686](#)
- commands, disabling CP [314](#)
- commands, enabling CP [461](#)
- configuration mode, displaying status [807](#)
- control unit
  - changing dynamically [691](#)
  - defining dynamically [166](#)
  - deleting dynamically [264](#)
- CP command, creating alias for [129](#)
- CP routines, loading [114](#)
- CP routines, unloading [118](#)
- CPCACHE command [95](#)
- CPCACHE FILES file [95](#)
- CPHX command [97](#)
- CPLISTFILE command [101](#)
- CPRELEASE command [103](#)
- creating alias for CP command [129](#)
- define a dedicated-reconfigurable PCI function [201](#)
- DEFINE COMMAND command [147](#)
- DEFINE CPOWNED command [151](#)
- DEFINE TIMEZONE command [226](#)
- defining
  - channel paths dynamically [142](#)
  - control units dynamically [166](#)
  - devices dynamically [170](#)
  - paths dynamically, channel [142](#)
  - real devices dynamically [170](#)
- delete PCI function [272](#)
- deleting
  - channel paths dynamically [260](#)
  - control blocks, real device [274](#)
  - control units dynamically [264](#)
  - device control blocks, real [274](#)
  - devices dynamically [266](#)
  - paths dynamically, channel [260](#)
  - RDEVs [274](#)
  - real device control blocks [274](#)
  - real devices dynamically [266](#)
  - units dynamically, control [264](#)
- device
  - control blocks, deleting real [274](#)
  - defining real, dynamically [170](#)
  - definition, changing real dynamically [695](#)
  - deleting real, dynamically [266](#)
- DIAGNOSE codes, disabling [317](#)
- DIAGNOSE codes, enabling [464](#)
- disabling CP commands [314](#)
- disabling DIAGNOSE codes [317](#)

example (continued)

- displaying
  - active partition [991](#)
  - address of CMDBK [606](#), [819](#)
  - address of CP command table entry block [606](#), [819](#)
  - address of CP exit block [650](#)
  - address of DGNBK [609](#)
  - address of DIAGNOSE code [609](#)
  - address of ICLBK [615](#)
  - address of indirect call locator block [615](#)
  - address of XITBK [650](#)
  - all partitions [991](#)
  - available storage for dynamic I/O changes [897](#)
  - call to external entry point, indirect [943](#)
  - channel path status, logical [792](#), [798](#), [1275](#)
  - channel path status, physical [796](#)
  - CHPID virtualization directory setting [798](#), [1275](#)
  - CMDBK address [606](#), [819](#)
  - configuration mode status [807](#)
  - CP command table entry block address [606](#), [819](#)
  - CP exit block address [650](#)
  - CP exit point information [908](#)
  - CP languages, valid [826](#)
  - CP routines information [849](#)
  - devices installed on logical channel path [792](#)
  - devices, whether CP initializes new [1033](#)
  - DGNBK address [609](#)
  - DIAGNOSE code address [609](#)
  - DIAGNOSE code information [886](#)
  - DIAGNOSE codes [175](#)
  - dynamic I/O status [896](#)
  - entry point, indirect call to external [943](#)
  - entry points for DIAGNOSE code [886](#)
  - entry points, references to unresolved [1250](#)
  - external entry point, indirect call to [943](#)
  - external symbols, references to unresolved [1250](#)
  - HSA storage for dynamic I/O changes [897](#)
  - I/O rates for throttled devices [1217](#)
  - ICLBK address [615](#)
  - identifiers, language [826](#)
  - if collecting subchannel measurement data [1150](#)
  - indirect call locator block address [615](#)
  - indirect call to external entry point [943](#)
  - information about CP routines [849](#)
  - IPL record contents [954](#)
  - language identifiers [826](#)
  - languages, valid CP [826](#)
  - load ID information [849](#)
  - logical channel path status [792](#), [798](#), [1275](#)
  - LPAR information [991](#)
  - system status [995](#)
  - message repository languages [827](#)
  - new devices, whether CP initializes [1033](#)
  - partitions [991](#)
  - path status, logical channel [792](#)
  - path status, physical channel [796](#)
  - physical channel path status [796](#)
  - privilege classes for DIAGNOSE code [886](#)
  - references to unresolved entry points [1250](#)
  - references to unresolved external symbols [1250](#)
  - status information for CP exit points [908](#)
  - status of DIAGNOSE code [886](#)
  - status, configuration mode [807](#)
  - status, dynamic I/O [896](#)

example (continued)

displaying (continued)

- status, logical channel path [792](#)
- status, lsystem [995](#)
- status, physical channel path [796](#)
- throttled devices [1217](#)
- total SCMBK space [1148](#)
- unresolved entry points, references to [1250](#)
- unresolved external symbols, references to [1250](#)
- unused SCMBK space [1148](#)
- usage information for CP exit points [908](#)
- valid CP languages [826](#)
- whether CP initializes new devices [1033](#)
- XITBK address [650](#)

displaying configuration mode status [807](#)

displaying lsystem status [995](#)

enabling CP commands [461](#)

enabling DIAGNOSE codes [464](#)

entry points, assigning to exit points [22](#)

exit point, start having CP use [466](#)

exit point, stopping CP from using [320](#)

exit points, assigning entry points to [22](#)

loading CP routines [114](#)

locate

PCIF [18 621](#)

RPCI [18 621](#)

VPCI \* [17 645](#)

LOCATE (Storage) command [632](#)

LOCATE FILID command [611](#)

LOCATE FRAMETBL command [612](#)

LOCATE LDEV command [616](#)

LOCATE RDEV command [619](#)

LOCATE SHPBK command [623](#)

LOCATE SNABK command [626](#)

LOCATE SPFBK command [629](#)

LOCATE SXSTE command [634](#)

LOCATE SYMBOL command [637](#)

LOCATE VDEV command [640](#)

LOCATE VMDBK command [643](#)

LOCATE VSMBK command [647](#)

LOCATEVM command [653](#)

locating CP command table entry block address [606](#), [819](#)

LOCK command [657](#)

lsystem, displaying status [995](#)

message repositories, using local [26](#)

modifying

channel paths dynamically [681](#)

control units dynamically [691](#)

device definitions dynamically, real [695](#)

paths dynamically, channel [681](#)

real device definitions dynamically [695](#)

overriding commands [686](#)

overriding DIAGNOSE codes [699](#)

path, channel

changing channel, dynamically [681](#)

defining channel, dynamically [142](#)

deleting channel, dynamically [260](#)

prompt from LOCATE (Storage) command [632](#)

prompt from LOCATEVM command [653](#)

QUERY CPDISKS command [824](#)

QUERY CPOWNED command [834](#)

QUERY CPSERVICE [839](#)

QUERY CTCA command [858](#)

example (continued)

QUERY D8ONECMD command [865](#)

QUERY FCP command [912](#)

QUERY HOTIO command [938](#)

QUERY LINES command [977](#)

QUERY LOGMSG command [990](#)

QUERY MDISK command [1005](#)

QUERY OSA command [1056](#)

QUERY SCMBKS command [1148](#)

QUERY SCMEASURE command [1150](#)

QUERY SWCH command [1195](#)

QUERY TAPES command [1208](#)

QUERY THROTTLE command [1217](#)

QUERY UR command [1253](#)

RDEVs, deleting [274](#)

real device

control blocks, deleting [274](#)

defining dynamically [170](#)

definition, changing dynamically [695](#)

deleting dynamically [266](#)

redefining commands [686](#)

redefining DIAGNOSE codes [699](#)

SET DEVICES [1491](#)

SET DYNAMIC\_I/O command [1502](#)

SET HOTIO command [1518](#)

set io\_opt [1522](#)

set ioassist [1524](#)

SET IOCD\_ACTIVE command [1526](#)

SET NEW\_DEVICES [1581](#)

SET RDEVICE commands

3800 printers [1669](#)

clear for all devices [1637](#)

for 3215 terminal [1662](#)

for advanced function printers [1630](#)

for afp device [1630](#), [1632](#), [1633](#)

for card punches [1632](#)

for card readers [1635](#)

for channel-to-channel adapter [1656](#)

for graphic display devices [1646](#)

for impact printers [1651](#)

for integrated communication adapters [1654](#)

for line adapters [1639](#)

for OSA device [1656](#)

for reader [1635](#)

for shared DASD [1642](#)

for tape units [1659](#)

for unsupported DASD [1664](#)

SET TRACEFRAMES command [1743](#)

start having CP use exit point [466](#)

status, displaying configuration mode [807](#)

status, displaying lsystem [995](#)

stopping CP from using exit point [320](#)

system execution space

loading CP routines into [114](#)

unloading CP routines from [118](#)

turning on or off

configuration mode [1437](#)

DIAGNOSE codes [317](#), [464](#)

dynamic I/O configuration [1502](#)

subchannel measuring [1686](#)

unloading CP routines [118](#)

using local message repositories [26](#)

execute

execute (*continued*)

CP commands using #CP [13](#)

exit

assigning entry points and external symbols to exit point [20](#)

defining [176](#)

disabling [319](#)

enabling [465](#)

entry point, assigning to exit point [20](#)

external symbol, assigning to exit point [20](#)

loading exit processing code into the system execution space [111](#)

modifying [700](#)

querying [907](#)

removing [700](#)

removing exit processing code from the system

execution space [117](#)

EXPLORE FCP command [467](#)

EXTERNAL command [469](#)

external symbol address

locating [636](#)

## F

FAIL operand

SET D8ONECMD command [1482](#)

FCB operand

SET RDEVICE command [1668](#)

START UR command [1870](#)

FCB option

CLOSE command [71](#)

FCP operand

SET RDEVICE command [1655](#)

file

causing CP to open [94](#)

checking system configuration [2143](#)

copies, increasing output [1384](#)

displaying selected lines of [105](#)

IOCDs

changing for next POR [1525](#)

logo configuration

adding to [1380](#)

checking [2143](#)

replacing [1380](#)

spool

block, locating [627](#)

class, specifying for impact printers [1649](#)

identification entry, locating [610](#)

printer [70](#)

punch [70](#)

reader [70](#)

file mode

establishing [89](#)

FILEFCB operand

START UR command [1871](#)

files

virtual reader

putting in order [2243](#)

find

CP command table entry block [603](#)

CP DIAGNOSE code table block [608](#)

data in host storage [630](#)

exit control block [649](#)

exit point indirect call locator block [614](#)

find (*continued*)

external symbol address [636](#)

shared system name block [623](#)

spool file block [627](#)

spool file identification entry [610](#)

system execution space page management table entry [634](#)

system frame table entry [612](#)

system logical device block [616](#)

system real device block [618](#)

system SNA control block [625](#)

system VTAM service machine block [647](#)

virtual device block [639](#)

virtual machine definition block [642](#)

FLASH operand

SET RDEVICE command [1668](#)

START UR command [1871](#)

FLASH option

CLOSE command [71](#)

FLASHCOPY [471](#)

FLASHCOPY BACKGNDCOPY command [480](#)

FLASHCOPY ESTABLISH command [483](#)

FLASHCOPY RESYNC command [489](#)

FLASHCOPY TGTWRITE command [492](#)

FLASHCOPY WITHDRAW command [494](#)

FLASHCOPY)

PRESERVEmirror [471](#), [483](#)

FLUSH command [498](#)

FOLD operand

START UR command [1871](#)

FOLDUP operand

SET RDEVICE command [1650](#)

force

single-spaced printer output [1804](#)

FORCE command [504](#)

FORCE operand

DISABLE HCD command [321](#)

FORCEOFF operand

SET CACHE command [1422](#)

SET DASDFW command [1485](#)

forcing hardstop with soft abends

using SET ABEND [1413](#)

FORM operand

SET RDEVICE command for 3800 printers [1668](#)

SET RDEVICE command for card punches [1632](#)

SET RDEVICE command for impact printers [1650](#)

START UR command [1869](#)

FORM option

FORM operand [71](#)

format

controller dump file [2114](#)

DASD volume for CP [2124](#)

FORWARD command [508](#)

frame allocation

for internal trace tables [1742](#)

for internal trace tables, querying [1230](#)

frame table

entry, locating [612](#)

free

spool files from system HOLD [511](#)

user from LOGON HOLD [510](#)

FREE LOGON command [510](#)

FREE operand

QUERY ALL command [759](#)

FREE operand (*continued*)

- QUERY CTCA command [857](#)
- QUERY DASD command [869](#)
- QUERY FCP command [912](#)
- QUERY GRAF command [932](#)
- QUERY LINES command [976](#)
- QUERY OSA command [1056](#)
- QUERY SWITCHES command [1194](#)
- QUERY TAPES command [1208](#)
- QUERY UR command [1252](#)
- FREE PRINTER/PUNCH command [511](#)
- freeing established cross-system links [2112](#)
- full-pack minidisks
  - displaying device characteristics [1004](#)
  - QUERY LKFACR command [983](#)
  - SET LKFACR command [1542](#)

## G

- GATEWAY operand
  - QUERY CONV command [810](#)
- generating I/O configuration data sets
  - using IOCP [2203](#)
- GENIMAGE utility [2192](#)
- GIVE (real device) command [515](#)
- GIVE (virtual device) command [518](#)
- GIVE command [513](#)
- GRAF operand
  - DEFINE GRAF command [179](#)
  - SET HOTIO command [1516](#)
- graphic display devices
  - changing [1645](#)
  - clearing RDEV information [1636](#)
  - defining [1645](#)
- guest operating system
  - messages, recalling [1349](#)

## H

- halt
  - active channel program [520](#)
  - all virtual processors (soft) [1878](#)
  - ATTACH commands [96](#)
  - channel program, active [520](#)
  - console sharing [1688](#)
  - event monitoring [727](#)
  - journal security function [1533](#)
  - LOCATE commands [96](#)
  - LOCATEVM commands [96](#)
  - monitoring of sample data collection [719](#)
  - new operations on real DASD [386](#)
  - processing tape files [1844](#)
  - program, active channel [520](#)
  - sample monitoring [727](#)
  - selected real processors (hard) [1878](#)
  - spool
    - activity on virtual spool devices [70](#)
    - file [498](#)
    - operations on real unit record devices [388](#)
  - system functions [1789](#)
  - tape procedures [1834](#)
  - use of files on minidisk accessed by CP [103](#)

halt (*continued*)

- VARY commands [96](#)
- virtual machine session [659](#)
- HALT command [520](#)
- HCD
  - disabling [321](#)
  - querying [934](#)
- HCPLDR utility
  - statements
    - conditional page boundary control [2272](#)
    - delete control [2276](#)
    - include control section control [2276](#)
    - loader control [2271](#)
    - loader termination control [2274](#)
    - padding control [2273](#)
    - parameter control [2273](#)
    - printer control [2271](#)
    - replace control [2274](#)
    - set location counter control [2272](#)
    - set page boundary control [2271](#)
    - subsystem control [2274](#)
    - unconditional page boundary control [2272](#)
    - verify control [2275](#)
- HELP, online [10](#)
- HEX operand
  - LOCATE (Storage) command [631](#)
  - LOCATEVM command [652](#)
- highlighting command redisplay
  - using TERMINAL HILIGHT [1931](#)
- HIPERSockets operand
  - QUERY OSA command [1056](#)
  - SET RDEVICE command [1655](#)
- hold
  - current output [498](#)
  - spool files [523](#)
- HOLD LOGON command [522](#)
- HOLD operand
  - SET RDEVICE command [1646](#)
  - START UR command [1872](#)
- HOLD option
  - CLOSE command [71](#)
- HOLD PRINTER/PUNCH command [523](#)
- host access list
  - displaying entries [335](#)
  - dumping entries [399](#)
- host system
  - accessing [662](#)
  - attaching DASD to [32](#)
  - collecting data for [707](#)
  - DETACH (Real Device) [296](#)
  - DETACH command [279](#)
  - disabling access to [311](#)
  - disabling logical connections [2053](#)
  - DISCONNECT [329](#)
  - displaying command limit threshold [799](#)
  - displaying operating load [535](#)
  - enabling device access to [458](#)
  - enabling logical connections [2053](#)
  - letting user access [510](#)
  - LOGOFF [659](#)
  - monitoring [707](#)
  - preventing access to [522](#)
  - reporting data about [707](#)
- hot I/O rate

hot I/O rate (*continued*)  
  changing [1516](#)  
  querying [938](#)  
  setting [1516](#)  
HSA storage  
  displaying amount available [895](#)  
HYPERSWAP  
  querying [941](#)  
HYPERSWAP command [525](#)

## I

I/O (Input/Output)  
  dynamic  
    commands, using in configuration mode [1436](#)  
    displaying status [895](#)  
    displaying storage available for changes [895](#)  
  machine check  
    initializing devices after [1581](#)  
  throttling  
    adding real devices [1734](#)  
    changing I/O rates [1734](#)  
    clearing I/O rates [1734](#)  
    deleting real devices [1734](#)  
    disabling [1734](#)  
    displaying status [1217](#)  
    enabling [1734](#)  
I/O configuration  
  adding  
    channel path [136](#)  
    channel path to control unit [688](#)  
    control unit [162](#)  
    control unit to real device [693](#)  
    device [168](#)  
    partition to channel path [679](#)  
    partition to real device [693](#)  
    real device [168](#)  
    unit address to control unit [688](#)  
  changing  
    channel path [679](#)  
    control unit [688](#)  
    device [693](#)  
    path, channel [679](#)  
    real device [693](#)  
  creating  
    channel path [136](#)  
    control unit [162](#)  
    device [168](#)  
    real device [168](#)  
  defining  
    channel path [136](#)  
    control unit [162](#)  
    device [168](#)  
    real device [168](#)  
  deleting  
    channel path [259](#)  
    channel path from control unit [688](#)  
    control unit [263](#)  
    control unit from real device [693](#)  
    devices [265](#)  
    partition from channel path [679](#)  
    partition from real device [693](#)  
    real devices [265](#)  
    unit address from control unit [688](#)

I/O configuration (*continued*)  
  modifying  
    channel path [679](#)  
    control unit [688](#)  
    device [693](#)  
    path, channel [679](#)  
    real device [693](#)  
I/O rate, hot  
  changing [1516](#)  
  querying [938](#)  
  setting [1516](#)  
IBM1 operand  
  DEFINE LINE command [186](#)  
IBM1\_ADAPTER operand  
  SET RDEVICE command [1638](#)  
IBMCLASS operand  
  LOCATE CMDBK command [603](#)  
  QUERY CPCMDS command [818](#)  
ICA\_ETHERNET operand  
  SET RDEVICE command [1653](#)  
ICKDSF, using to format DASD [2129](#)  
identification entry, spool file, locating [610](#)  
identify  
  authorized commands [802](#)  
  authorized DIAGNOSE codes [802](#)  
  communication link to ISFC [16](#)  
  CP commands issued [88](#)  
  minidisk to CP [89](#)  
  valid CP languages [826](#)  
IGNORE operand  
  LOCATE (Storage) command [631](#)  
  LOCATEVM command [652](#)  
image library  
  dumping files to tape [1836](#)  
  files, purging [743](#)  
  loading files from tape [1846](#)  
  LOADVFCB [600](#)  
IMAGE operand  
  START UR command [1871](#)  
IMAGE\_LIBRARY operand  
  SET RDEVICE command [1650](#), [1669](#)  
IMAGELIB utility [2201](#)  
impact printer  
  changing [1648](#)  
  clearing RDEV information [1636](#)  
  defining [1648](#)  
IMPACT\_PRINTER operand  
  SET RDEVICE command [1649](#)  
increase  
  output file copies [1384](#)  
INCREMENT operand  
  LOCATE (Storage) command [631](#)  
  LOCATEVM command [652](#)  
INDEX operand  
  START UR command [1871](#)  
INDICATE ACTIVE command [532](#)  
INDICATE command [531](#)  
INDICATE I/O command [534](#)  
INDICATE LOAD command [535](#)  
INDICATE MULTITHREAD command [539](#)  
INDICATE NSS command [543](#)  
INDICATE PAGING command [546](#)  
INDICATE QUEUES command [548](#)  
INDICATE SPACES command [552](#)

INDICATE USER command [555](#)  
indirect addressing [333](#), [397](#)  
INFORMATION operand  
  CPLISTFILE command [100](#)  
initialize  
  CSE area on DASD volume [2260](#)  
  devices  
    after IOMCK [1581](#)  
    when added to the system [1581](#)  
input/output configuration (IOCP) utility [705](#)  
install  
  copy of Stand-Alone Program Loader  
  [2222](#)  
integrated communications adapters  
  changing [1653](#)  
  clearing RDEV information [1636](#)  
  defining [1653](#)  
intercepting machine check interrupts  
  using TRACE MCH [1985](#)  
internal trace table  
  allocating space [1742](#)  
  displaying space allocated for [1230](#)  
interrupt  
  checks, controlling frequency of [1571](#)  
  clearing pending [1387](#)  
  creating pending attention [1386](#)  
  virtual machine operation [49](#)  
  VM execution using #CP [13](#)  
interrupt, unsolicited  
  changing [1516](#)  
  querying [938](#)  
  setting [1516](#)  
interval  
  timer, controlling [1737](#)  
invoking EREP program  
  using CPEREPXA [2118](#)  
IOCP utility [2203](#)  
IOEXPLOR utility [2205](#)  
**IPL** command [566](#)  
IPLPARMS operand  
  SALIPL command [2225](#)  
IPWIZARD utility [2210](#)  
ISFC (Inter-System Facility for  
  Communications)  
  identifying communication link [16](#)  
IUCV operand  
  SET EMSG command [1512](#)  
  SET IMSG command [1519](#)  
  SET MSG command [1576](#)  
  SET SMSG command [1703](#)  
  SET VMCONIO command [1751](#)  
  SET WNG command [1784](#)

## K

KEEP option  
  CLOSE command [72](#)  
key  
  PF (Program Function)  
    assigning [1602](#)  
    displaying retrieve buffers [1143](#)  
    RETRIEVE buffers [1681](#)

## L

label  
  DASD volume for CP [2124](#)  
language  
  identifiers, displaying [826](#)  
  identifying which used for CP messages [828](#)  
  valid CP, displaying [826](#)  
limit  
  incorrect passwords [1432](#)  
LIMIT operand  
  RECORDING command [1375](#)  
  SET RECORD command [1671](#)  
  START UR command [1869](#)  
line adapters  
  changing [1638](#)  
  clearing RDEV information [1636](#)  
  defining [1638](#)  
lines of file, displaying [105](#)  
lines, deleting at terminal before entering [2267](#)  
LINK command [578](#)  
LINKS operand  
  START (disk) command [1864](#)  
list  
  files  
    on minidisk accessed by CP [99](#)  
  real device number of most recent I/O map  
  [534](#)  
  virtual machines in I/O wait state [534](#)  
  virtual machines in page wait status [546](#)  
load  
  communication controller [2116](#)  
  CP routines into the system execution space [111](#)  
  FCB buffer [592](#)  
  FCB image [598](#)  
  files from tape [1846](#)  
  modules with Stand-Alone Program Loader  
  [2222](#)  
  UCS buffer [592](#)  
LOADADDR operand  
  SALIPL command [2223](#)  
LOADBUF command [592](#)  
loader, controlling system [2271](#)  
LOADVFCB command [598](#)  
local modifications  
  displaying [839](#)  
locate  
  CP command table entry block [603](#)  
  CP DIAGNOSE code table block [608](#)  
  data in host storage [630](#)  
  exit control block [649](#)  
  exit point indirect call locator block [614](#)  
  external symbol address [636](#)  
  shared system name block [623](#)  
  spool file block [627](#)  
  spool file identification entry [610](#)  
  system execution space page management table entry  
  [634](#)  
  system frame table entry [612](#)  
  system logical device block [616](#)  
  system real device block [618](#)  
  system SNA control block [625](#)  
  system VTAM service machine block [647](#)  
  virtual device block [639](#)

locate (*continued*)  
 virtual machine definition block [642](#)  
 LOCATE CMDBK command [603](#)  
 LOCATE command [602](#)  
 LOCATE DGNBK command [608](#)  
 LOCATE FILID command [610](#)  
 LOCATE FRAMETBL command [612](#)  
 LOCATE FRMTBL command [612](#)  
 LOCATE FRMTE command [612](#)  
 LOCATE ICLBK command [614](#)  
 LOCATE LDEV command [616](#)  
 LOCATE RDEV command [618](#)  
 LOCATE RPCI command [621](#)  
 LOCATE SHPBK command [623](#)  
 LOCATE SNABK command [625](#)  
 LOCATE SPFBK command [627](#)  
 LOCATE Storage command  
 command description [630](#)  
 stopping [96](#)  
 LOCATE SXSTE command [634](#)  
 LOCATE SYMBOL command [636](#)  
 LOCATE VDEV command [639](#)  
 LOCATE VMDBK command [642](#)  
 LOCATE VPCI command [645](#)  
 LOCATE VSMBK command [647](#)  
 LOCATE XITBK command [649](#)  
 LOCATEVM command  
 command description [651](#)  
 stopping [96](#)

location  
 of CP module, displaying [832](#)  
 of CP nucleus, displaying [832](#)  
 of parm disk, displaying [832](#)  
 storing data at specified (ESA/XC, z/XC) [1881](#)

LOCATION operand  
 QUERY MDISK command [1004](#)

LOCK command [654](#)

LOCK operand  
 SET D8ONECMD command [1482](#)  
 SET PRIVCLASS command [1616](#)

lock pages of guest storage to host storage [654](#)

log on  
 defining new CP command to issue before [143](#)  
 to another virtual machine using AUTOLOG [50](#)

LOG operand  
 QUERY HOLD command [936](#)  
 SET D8ONECMD command [1482](#)

log, messages  
 displaying [988](#)  
 migration note [988](#)

logging to another virtual machine  
 using XAUTOLOG [2100](#)

logical device  
 attaching to host system [32](#)

logical line editing symbols [2267](#)

logical partition (LPAR)  
 active, displaying [991](#)  
 all, displaying [991](#)  
 displaying [991](#)

LOGIN command [662](#)

logo configuration file  
 adding to [1380](#)  
 replacing [1380](#)

LOGOFF command [659](#)

LOGON command [662](#)

LOGOUT command [659](#)

LOGROUTE parameter of the RETRIEVE utility [2219](#)

## M

machine check  
 I/O  
 initializing devices after [1581](#)

magnetic tape drives, displaying status [1207](#)

MANUAL operand  
 START UR command [1869](#)

MAP operand  
 LOCK command [655](#)  
 QUERY ALLOC command [773](#)  
 QUERY NSS command [1042](#)

MASTER operand  
 SET TRACEFRAMES command [1742](#)

maximum  
 number of PF key RETRIEVE buffers, setting [1681](#)

MAXIMUM operand  
 SET RETRIEVE command [1681](#)

maximum user limit [51](#), [999](#), [1559](#)

MDC operand  
 RDEVICE statement [1642](#)

MDR (Miscellaneous Data Record), retrieving [1910](#)

MEMBER operand  
 CPLISTFILE command [100](#)  
 CPTYPE command [105](#)  
 CPXLOAD command [112](#)

merging trace traps into one trace set  
 using TRACE APPEND [1965](#)

message  
 controlling display of warning [1784](#)  
 controlling receipt of special [1703](#)

log on  
 displaying [988](#)  
 recalling guest operating system [1349](#)  
 sending [673](#), [676](#)  
 sending special [1800](#)  
 sending to all active users [674](#)

MESSAGE (user) command [676](#)

MESSAGE ALL command [674](#)

MESSAGE command [673](#)

message examples, notation used in [6](#)

message repository  
 files, purging [745](#)

MIGR51D utility [2211](#)

migration note about system log messages [988](#)

minidisk  
 accessing  
 by CP [89](#)  
 adding [188](#)  
 adding temporary [223](#)  
 defining [188](#)  
 defining temporary [223](#)  
 displaying device characteristics [1004](#)  
 files  
 displaying selected lines of [105](#)  
 listing information about [99](#)  
 opening file on a [94](#)  
 releasing [103](#)

MINIVOL operand  
 SALIPL command [2223](#)



Miscellaneous Data Record (MDR), retrieving [1910](#)  
 MODEL operand  
   SET RDEVICE command [1667](#)  
 modify  
   3800 printers [1667](#)  
   advanced function printers [1629](#)  
   card  
     punches [1631](#)  
     readers [1634](#)  
   channel path [679](#)  
   channel-to-channel adapter [1655](#)  
   communication controllers [1638](#)  
   control unit [688](#)  
   CP-owned volume list [150](#)  
   CTCA [1655](#)  
   D8ONECMD settings [1482](#)  
   DASD [1641](#)  
   defaults  
     for Stand-Alone Program Loader  
       [2222](#)  
   device [693](#)  
   device numbers, virtual [227](#), [1378](#)  
   emulated devices [1504](#)  
   event data collection profile [711](#)  
   exit [700](#)  
   external attributes of closed spool files [60](#)  
   FCP device [1655](#)  
   graphic display devices [1645](#)  
   HiperSockets device [1655](#)  
   hot I/O rate [938](#), [1516](#)  
   I/O configuration [122](#)  
   impact printers [1648](#)  
   integrated communication adapters [1653](#)  
   IOCDS file for next POR [1525](#)  
   IODC characteristics of real device [693](#)  
   line adapters [1638](#)  
   number of  
     frames for internal trace tables [1742](#)  
     PF key RETRIEVE buffers [1681](#)  
   OSA device [1655](#)  
   path, channel [679](#)  
   privilege class [1616](#)  
   processing options for real DASD [1864](#)  
   PSW data [1901](#)  
   PSWA data [1903](#)  
   queued spool file order [735](#)  
   real device [693](#)  
   real PCI function [703](#)  
   real storage [1879](#)  
   record processing parameters [1375](#)  
   sample data collection profile [719](#)  
   secondary user [1688](#)  
   session account number [1416](#)  
   special devices [1655](#)  
   spool options [1808](#)  
   status indicator [734](#)  
   storage size [213](#)  
   system parameters [1705](#)  
   system-resource-access-priority for users [1691](#)  
   system's definition of a set of real devices [1627](#)  
   tape units [1658](#)  
   teleprocessing integrated adapters [1653](#)  
   time zone [225](#), [1738](#)  
   trace traps [1973](#), [2002](#)

modify (*continued*)  
   unsolicited interrupt rate [938](#), [1516](#)  
   unsupported devices [1663](#)  
   user  
     HOLD status of active spool files [60](#)  
     ID of primary system operator [1731](#)  
   virtual  
     configuration [122](#)  
     device numbers [1378](#)  
     machine language [1441](#)  
     processor address [152](#)  
     processor configuration type [1555](#)  
     processor ID [1474](#)  
     virtual machine [1591](#)  
     virtual storage [1879](#)  
 MODIFY CHPID command [679](#)  
 MODIFY CMD command [683](#)  
 MODIFY CNTLUNIT command [688](#)  
 MODIFY COMMAND command [683](#)  
 MODIFY CU command [688](#)  
 MODIFY DEVICE command [693](#)  
 MODIFY DIAGNOSE command [697](#)  
 MODIFY EXIT command [700](#)  
 MODIFY IODEVICE command [693](#)  
 MODIFY option  
   CLOSE command [72](#)  
 MODIFY PATH command [679](#)  
 MODIFY PCIFUNCTION command [703](#)  
 MODIFY PRIV\_CLASSES command [705](#)  
 module  
   CP  
     displaying location of [832](#)  
 monitor  
   command [707](#)  
   displaying profile [1014](#)  
   events [727](#)  
   record data, controlling [1575](#)  
   samples [727](#)  
   system data [707](#)  
   terminal event records [1013](#)  
   virtual machine events [1076](#)  
   VM events [739](#)  
 MONITOR  
   communication controller for interrupts [2116](#)  
   TRACE SVC/DIAGNOSE/MC [1998](#)  
   TRSOURCE ID (Definition Operands) [2030](#)  
 MONITOR command [707](#)  
 MONITOR EVENT command [711](#)  
 MONITOR SAMPLE command [719](#)  
 MONITOR START command [727](#)  
 MONITOR STOP command [727](#)  
 monitoring virtual machine events  
   using TRACE [1946](#)  
 MONWRITE utility [2212](#)  
 MONWSTOP utility [2216](#)  
 MOVE2SFS utility [2217](#)  
 moving components to SFS directories [2217](#)  
 MSG option  
   CLOSE command [72](#)  
 MSGNOH command [731](#)  
 MSGProc [191](#)  
 multithreading status, displaying [539](#), [1026](#), [1578](#)

## N

NAME operand  
  QUERY IMG command [945](#)  
  QUERY NLS command [1038](#)  
  QUERY NSS command [1041](#)  
  QUERY UCR command [1246](#)  
  SALIPL command [2223](#)

NAME option  
  CLOSE command [72](#)

name table, shared system, locating [623](#)

new  
  alias  
    creating for existing CP command [122](#), [127](#)  
  devices  
    displaying whether CP initializes [1033](#)  
  DIAGNOSE codes  
    creating [122](#), [172](#)  
  operations on real DASD, stopping [386](#)  
  version  
    creating for existing CP command [122](#), [143](#)

NO\_SPOOLING operand  
  SET RDEVICE command [1632](#), [1634](#), [1649](#), [1667](#)

NO3800 operand  
  START UR command [1870](#)

NOAFP operand  
  START UR operand [1870](#)

NOASSIGN operand  
  ATTACH command [34](#)

NOCTL operand  
  ATTACH command [35](#)

NODATCK operand  
  DEFINE (spooling device) command [209](#)  
  QUERY VIRTUAL PRINTER command [1324](#)

NODE operand  
  ACTIVATE ISLINK command [16](#)

NODEID operand  
  QUERY CONV command [810](#)  
  QUERY GATEWAY command [929](#)  
  QUERY RESOURCE command [1138](#)

NOFOLD operand  
  START UR command [1871](#)

NOFORCE operand  
  DISABLE HCD command [321](#)

NOHOLD option  
  CLOSE command [71](#)

NOKEEP option  
  CLOSE command [72](#)

NOMAP operand  
  LOCK command [655](#)

NOMSG option  
  CLOSE command [72](#)

NONAME option  
  CLOSE command [72](#)

nonvolatile storage (NVS) [277](#)

NOSEP operand  
  START UR command [1869](#)

notation used in message and response examples [6](#)

NOTREADY command [734](#)

NOVERIFY operand  
  SALIPL command [2223](#), [2224](#)

nucleus  
  CP  
    displaying location of [832](#)

## O

OFF operand  
  RECORDING command [1375](#)  
  SET 370ACCOM command [1787](#)  
  SET CACHEFW command [1427](#)  
  SET D8ONECMD command [1482](#)  
  SET DASDFW command [1485](#)  
  SET EMSG command [1512](#)  
  SET HOTIO command [1517](#)  
  SET IMSG command [1519](#)  
  SET LINEDIT command [1539](#)  
  SET MAXUSERS command [1559](#)  
  SET MDCACHE command [1563](#)  
  SET MIH command [1570](#)  
  SET MONDATA command [1575](#)  
  SET MSG command [1576](#)  
  SET NOPDATA command [1587](#)  
  SET NVS command [1588](#)  
  SET OBSERVER command [1591](#)  
  SET PAGEX command [1594](#)  
  SET QUICKDSP command [1625](#)  
  SET RECORD command [1671](#)  
  SET REORDER command [1672](#)  
  SET RESERVED command [1674](#)  
  SET RUN command [1683](#)  
  SET SECUSER command [1688](#)  
  SET SHUTSIGNAL command [1700](#)  
  SET SMSG command [1703](#)  
  SET STGEXEMPT command [1719](#)  
  SET STGLIMIT command [1721](#)  
  SET TIMER command [1737](#)  
  SET VMCONIO command [1751](#)  
  SET WNG command [1784](#)  
  SET WRKALLEG command [1785](#)  
  TERMINAL ATTN command [1921](#)  
  TERMINAL CHARDEL command [1926](#)  
  TERMINAL ESCAPE command [1930](#)  
  TERMINAL HOLD command [1932](#)  
  TERMINAL LINEDEL command [1933](#)  
  TERMINAL LINEND command [1934](#)  
  TERMINAL LINESIZE command [1935](#)  
  TERMINAL TABCHAR command [1942](#)

OFFLINE operand  
  QUERY ALL command [759](#)  
  QUERY CTCA command [857](#)  
  QUERY DASD command [869](#)  
  QUERY FCP command [912](#)  
  QUERY GRAF command [932](#)  
  QUERY LINES command [976](#)  
  QUERY OSA command [1056](#)  
  QUERY SWITCHES command [1194](#)  
  QUERY TAPES command [1208](#)  
  QUERY UR command [1252](#)

ON operand  
  RECORDING command [1375](#)  
  SET 370ACCOM command [1787](#)  
  SET CACHEFW command [1427](#)  
  SET DASDFW command [1485](#)  
  SET EMSG command [1512](#)  
  SET IMSG command [1519](#)  
  SET LINEDIT command [1539](#)  
  SET MDCACHE command [1563](#)  
  SET MIH command [1570](#)

ON operand (*continued*)  
 SET MONDATA command [1575](#)  
 SET MSG command [1576](#)  
 SET NOPDATA command [1587](#)  
 SET NVS command [1588](#)  
 SET PAGEX command [1594](#)  
 SET QUICKDSP command [1625](#)  
 SET RECORD command [1671](#)  
 SET REORDER command [1672](#)  
 SET RUN command [1683](#)  
 SET SHUTSIGNAL command [1700](#)  
 SET SMSG command [1703](#)  
 SET STGEXEMPT command [1719](#)  
 SET STGLIMIT command [1721](#)  
 SET TIMER command [1737](#)  
 SET WNG command [1784](#)  
 SET WRKALLEG command [1785](#)  
 TERMINAL ATTN command [1921](#)  
 TERMINAL CHARDEL command [1926](#)  
 TERMINAL ESCAPE command [1930](#)  
 TERMINAL HOLD command [1932](#)  
 TERMINAL LINEDEL command [1933](#)  
 TERMINAL LINEND command [1934](#)  
 TERMINAL TABCHAR command [1942](#)  
 online HELP Facility, using [10](#)  
 open  
   file on a minidisk [94](#)  
 OPER\_IDENT\_READER operand  
   SET RDEVICE command [1646](#)  
 operator  
   system  
     changing user ID of [1731](#)  
     displaying user ID of [1202](#)  
 option  
   displaying terminal [1213](#)  
 order  
   queued spool files [735](#)  
 ORDER command [735](#)  
 ORIGIN operand  
   QUERY LOGMSG command [989](#)  
 OSA (Open Systems Adapter)  
   displaying status [1055](#)  
 OSA operand  
   QUERY OSA command [1056](#)  
   SET RDEVICE command [1655](#)  
 OSN operand  
   QUERY OSA command [1056](#)  
 output  
   destination, controlling CP [1440](#)  
   file copies, increasing [1384](#)  
 OWN operand  
   DEFINE CPOWNER command [150](#)

## P

page  
   locking [654](#)  
   reserved, setting [1674](#)  
   unlocking [2043](#)  
 page allocation  
   displaying [773](#)  
 PAGE operand  
   DRAIN (disk) command [386](#)  
   QUERY ALLOC command [773](#)

PAGE operand (*continued*)  
   START (disk) command [1864](#)  
 Parallel Access Volume DASD status, displaying  
   QUERY DASD DETAILS command [867](#)  
   QUERY PAV command [1068](#)  
 parm disk  
   displaying location of [832](#)  
 password  
   AUTOLOG command [50](#)  
   displaying incorrect limit threshold [799](#)  
   displaying suppression status [1062](#)  
 path  
   installed to a device, displaying [1064](#)  
 path, channel  
   adding  
     to control unit [688](#)  
   changing [679](#)  
   creating [136](#)  
   defining [136](#)  
   deleting  
     from control unit [688](#)  
   disabling [2046](#)  
   displaying  
     devices for logical [791](#), [798](#), [1275](#)  
     status of logical [791](#), [798](#), [1275](#)  
     status of physical [796](#)  
     storage available for adding [895](#)  
   enabling [2046](#)  
 PCH operand  
   DRAIN UR command [388](#)  
   LOCATE SPFBK command [628](#)  
 pending interrupt, clearing [1387](#)  
 PER command [739](#)  
 PERCENT operand  
   SET TRACEFRAMES command [1742](#)  
 PERM option  
   DRAIN (disk) command [386](#)  
 PF (Program Function) keys  
   displaying  
     retrieve buffers [1143](#)  
     displaying assignments [1077](#)  
     RETRIEVE buffers [1681](#)  
 power-on reset (POR)  
   changing IOCDS file used at next [1525](#)  
 preface [xxv](#)  
 prevent  
   CP from allocating space for spool records [386](#)  
   new operations on real DASD [386](#)  
   TDISK requests [386](#)  
 print  
   controller dump file [2114](#)  
   user minidisks [2149](#)  
 printer  
   3800  
     changing [1667](#)  
     clearing RDEV information [1636](#)  
     defining [1667](#)  
   advanced function  
     changing [1629](#)  
     clearing RDEV information [1636](#)  
     defining [1629](#)  
   changing options [1808](#)  
   displaying status [1252](#)  
   forcing single-spaced output [1804](#)

printer (*continued*)

- impact
  - changing [1648](#)
  - clearing RDEV information [1636](#)
  - defining [1648](#)
- overriding carriage control commands [1804](#)
- print server virtual machine, displaying status of [1089](#)
- spool file
  - ending activity [70](#)
- stopping spool operations [388](#)
- unsupported
  - changing [1663](#)
  - clearing RDEV information [1636](#)
  - defining [1663](#)

PRINTER operand

- CLOSE command [70](#)
- DEFINE (spooling device) command [209](#)
- DRAIN UR command [388](#)
- LOCATE SPFBK command [628](#)
- SET RDEVICE command [1664](#)

PRINTER option

- CLOSE command [73](#)

privilege class

- changing [1616](#)
- displaying [1090](#)

privilege classes for users [6](#)

processor

- configuration mode, displaying setting [806](#)
- displaying
  - configuration mode setting [806](#)
  - displaying configuration mode setting [806](#)

processor core utilization information, displaying [539](#)

processor, adding or removing

- using VARY CORE command [2048](#)
- using VARY PROCESSOR command [2061](#)

producing configuration reports from existing IOCDs

- using IOCP [2203](#)

profile

- displaying monitor [1014](#)
- event data collection, changing [711](#)
- event data collection, setting [711](#)
- sample data collection, changing [719](#)
- sample data collection, setting [719](#)

PROMPT operand

- LOCATE (Storage) command [631](#)
- LOCATEVM command [652](#)

PRT operand

- DRAIN UR command [388](#)
- LOCATE SPFBK command [628](#)

PTFs

- displaying [839](#)

PUN operand

- DRAIN UR command [388](#)
- LOCATE SPFBK command [628](#)
- SET RDEVICE command [1631](#)

punch

- changing options [1808](#)
- displaying status [1252](#)
- spool file
  - ending activity [70](#)
- stopping spool operations [388](#)
- unsupported
  - changing [1663](#)
  - clearing RDEV information [1636](#)

punch (*continued*)

- unsupported (*continued*)
  - defining [1663](#)

PUNCH operand

- CLOSE command [70](#)
- DEFINE (spooling device) command [210](#)
- SET RDEVICE command [1664](#)

PUNCH option

- CLOSE command [73](#)

purge

- closed spool files [740](#)
- image library files [743](#)
- message repository files [745](#)
- NSS files [747](#)
- spool file [498](#)
- system data files [747](#)
- system trace files [749](#)
- UCR files [751](#)

PURGE command [740](#)

PURGE IMG command [743](#)

PURGE NLS command [745](#)

PURGE NSS command [747](#)

PURGE operand

- CLOSE command [72](#)
- RECORDING command [1376](#)
- START UR command [1872](#)

PURGE TRFILES command [749](#)

PURGE UCR command [751](#)

## Q

QID operand

- RECORDING command [1376](#)

query

- access modes to search for system logmsg [988](#)
- active partition [991](#)
- address
  - control information [602](#)
  - CP command table entry block [603](#), [817](#)
- all
  - partitions [991](#)
  - paths installed to a device [1064](#)
- buffers, PF key retrieve [1143](#)
- channel-to-channel adapters [857](#)
- communication lines [976](#)
- CP-owned volume list [834](#)
- current time zone definitions [1222](#)
- cylinder allocation [773](#)
- D8ONECMD settings [865](#)
- DASD [867](#)
- default reorder setting [1134](#)
- display devices [931](#)
- display status of real PCI function [1070](#), [1322](#)
- dynamic switching devices [1194](#)
- FCP [911](#)
- HCD status [934](#)
- hot I/O rate [938](#)
- HYPERSWAP status [941](#)
- interrupt rate, unsolicited [938](#)
- IPL parameters, system [954](#)
- key retrieve buffers, PF [1143](#)
- location
  - control information [602](#)
  - CP command table entry block [603](#), [817](#)

query (*continued*)

- location (*continued*)
  - CP module [832](#)
  - CP nucleus [832](#)
  - parm disk [832](#)
- log message [988](#)
- minidisk device characteristics [1004](#)
- Open Systems Adapters [1055](#)
- page allocation [773](#)
- partitions [991](#)
- PF key
  - retrieve buffers [1143](#)
- privilege class [1090](#)
- relocation domain [203](#), [1132](#), [1348](#)
- retrieve buffers, PF key [1143](#)
- server time protocol [1191](#)
- service [839](#)
- space allocated for internal trace tables [1230](#)
- spool file information [1104](#)
- status
  - all devices [759](#)
  - dynamic I/O support [895](#)
  - host real storage [759](#)
  - installed device paths [1064](#)
  - journaling, security [963](#)
  - logical devices [973](#)
  - measuring, subchannel [1149](#)
  - Parallel Access Volume DASD [867](#), [1068](#)
  - print server virtual machine [1089](#)
  - real channel-to-channel adapters [857](#)
  - real communication lines [976](#)
  - real DASD [867](#)
  - real devices [1114](#)
  - real display devices [931](#)
  - real dynamic switching devices [1194](#)
  - real magnetic tape drives [1207](#)
  - security journaling [963](#)
  - subchannel measuring [1149](#)
  - unit record devices [1252](#)
- storage
  - available for dynamic I/O changes [895](#)
- system operator [1202](#)
- system scheduler/dispatcher [531](#)
- unsolicited interrupt rate [938](#)
- user ID of system operator [1202](#)
- QUERY (Logical Device) command [973](#)
- QUERY (printer ID) command [1089](#)
- QUERY (real device) command [1114](#)
- QUERY (real) STORAGE command [1186](#)
- QUERY (virtual device) command [1291](#)
- QUERY ABEND command
  - to display ABEND SOFT status [753](#)
- QUERY ACCOUNT command [755](#)
- QUERY AGELIST command [757](#)
- QUERY ALL command [759](#)
- QUERY ALLOC command [773](#)
- QUERY BYUSER command [781](#)
- QUERY CACHE command [782](#)
- QUERY CACHEFW command [784](#), [786](#)
- QUERY CFLINKS [788](#)
- QUERY CHPID command [791](#)
- QUERY CHPIDS command [796](#)
- QUERY CHPIDV command [798](#)
- QUERY CMDLIMIT command [799](#)
- QUERY COLLECT command [800](#)
- QUERY COMMANDS command [802](#)
- QUERY CONCOPY command [804](#)
- QUERY CONFIGMODE command [806](#)
- QUERY CONTROLLER command [808](#)
- QUERY CONV command [810](#)
- QUERY CPCHECKING command [816](#)
- QUERY CPCMDS command [817](#)
- QUERY CPDISKS command [824](#)
- QUERY CPLANGLIST command [826](#)
- QUERY CPLANGUAGE command [828](#)
- QUERY CPLEVEL command [829](#)
- QUERY CPLOAD command [832](#)
- QUERY CPOWNED command [834](#)
- QUERY CPPROTECT command [836](#)
- QUERY CPSERVICE command [839](#)
- QUERY CPTRACE command [842](#)
- QUERY CPUAFFINITY command [845](#)
- QUERY CPUID command [846](#)
- QUERY CPUPOOL command superseded by QUERY RESPOOL command [1140](#)
- QUERY CPXLOAD command [848](#)
- QUERY CRYPTO command [853](#)
- QUERY CTCA command [857](#)
- QUERY CU command [860](#)
- QUERY D8ONECMD command [865](#)
- QUERY DASD command [867](#)
- QUERY DASD RESERVE command [867](#)
- QUERY DASDFW command [879](#)
- QUERY DATEFORMAT command [881](#)
- QUERY DEVICES command [883](#)
- QUERY DIAGNOSE command [885](#)
- QUERY DUMP command [887](#)
- QUERY DUMPDEV command [890](#)
- QUERY DUPLEX command [893](#)
- QUERY DYNAMIC\_I/O command [895](#)
- QUERY EDEVICE command [898](#)
- QUERY ENCRYPT command [904](#)
- QUERY EQID command [905](#)
- QUERY EXITS command [907](#)
- QUERY FCP command [911](#)
- QUERY FENCES command [915](#)
- QUERY FILES command [917](#)
- QUERY FLASHCOPY command [920](#)
- QUERY FRAMES command [925](#)
- QUERY GATEWAY command [929](#)
- QUERY GRAF command [931](#)
- QUERY HCD command [934](#)
- QUERY HOLD command [936](#)
- QUERY HOTIO command [938](#)
- QUERY HYPERSWAP command [941](#)
- QUERY ICLNAME command [943](#)
- QUERY IMG command [945](#)
- QUERY IO\_OPT command [948](#)
- QUERY IOASSIST command [950](#)
- QUERY IPLPARMS command [954](#)
- QUERY ISFC command [956](#)
- QUERY ISLINK command [959](#)
- QUERY JOURNAL command [963](#)
- QUERY KEYALIAS command [964](#)
- QUERY LAN command [966](#)
- QUERY LDEVS command [973](#)
- QUERY LINES command [976](#)

QUERY LINKS command [979](#)  
 QUERY LKFAC command [981](#)  
 QUERY LKFACR command [983](#)  
 QUERY LOADDEV command [985](#)  
 QUERY LOGMSG command [988](#)  
 QUERY LPARS command [991](#)  
 QUERY LSYSTEM command [995](#)  
 QUERY MAXLDEV command [996](#)  
 QUERY MAXSPOOL command [997](#)  
 QUERY MAXUSERS command [999](#)  
 QUERY MDCACHE command [1000](#)  
 QUERY MDISK command [1004](#)  
 QUERY MEMASSIST command [1008](#)  
 QUERY MITIME command [1011](#)  
 QUERY MONDATA command [1013](#)  
 QUERY MONITOR command [1014](#)  
 QUERY MSS command [1025](#)  
 QUERY MULTITHREAD command [1026](#)  
 QUERY NAMES commands [1029](#)  
 QUERY NEW\_DEVICES commands [1033](#)  
 QUERY NIC  
     examples [1034](#)  
 QUERY NIC command [1034](#)  
 QUERY NLS command [1038](#)  
 QUERY NSS command [1041](#)  
 QUERY NVS command [1050](#)  
 QUERY OBSERVER command [1052](#)  
 QUERY operand  
     LOCATE CMDDBK command [603](#)  
     QUERY CPCMDBS command [817](#)  
 QUERY OSA command [1055](#)  
 QUERY PAGING command [1060](#)  
 QUERY PASSWORD command [1062](#)  
 QUERY PATHS command [1064](#)  
 QUERY PAV command [1068](#)  
 QUERY PCIFUNCTION command [1070](#)  
 QUERY PENDING COMMANDS command [1074](#)  
 QUERY PER command [1076](#)  
 QUERY PFnn command [1077](#)  
 QUERY PINNED command [1079](#)  
 QUERY PORT command [1082](#)  
 QUERY PRINTER command [1104](#)  
 QUERY PRIVCLASS command [1090](#)  
 QUERY PROCESSORS command [1092](#)  
 QUERY PRODUCT command [1095](#)  
 QUERY PROMPT command [1097](#)  
 QUERY PSWTRans command [1099](#)  
 QUERY PUNCH command [1104](#)  
 QUERY PVMSG command [1349](#)  
 QUERY QIOASSIST command [1100](#)  
 QUERY QUICKDSP command [1102](#)  
 QUERY READER command [1104](#)  
 QUERY RECORDING command [1130](#)  
 QUERY RELODOMAIN command [1132](#)  
 QUERY RESERVED command [1135](#)  
 QUERY RESOURCE command [1138](#)  
 QUERY RESPOOL command [1140](#)  
 QUERY RETRIEVE command [1143](#)  
 QUERY RSAW command [1145](#)  
 QUERY SCMBKS command [1147](#)  
 QUERY SCMEASURE command [1149](#)  
 QUERY SCREEN command [1151](#)  
 QUERY SDF command [1152](#)  
 QUERY SECUSER command [1153](#)  
 QUERY SET command [1156](#)  
 QUERY SHARE command [1161](#)  
 QUERY SHUTDOWN command [1163](#)  
 QUERY SHUTDOWNTIME command [1166](#)  
 QUERY SIGNALS command [1168](#)  
 QUERY SPACES command [1170](#)  
 QUERY SRM command [1173](#)  
 QUERY SSI [1177](#)  
 QUERY STGEXEMPT command [1183](#)  
 QUERY STGLIMIT command [1185](#)  
 QUERY SUBSTITUTE command [1193](#)  
 QUERY SWCH command [1194](#)  
 QUERY SWITCH command [1194](#)  
 QUERY SXSPAGES command [1196](#)  
 QUERY SXSSTORAGE command [1199](#)  
 QUERY SYSASCII command [1200](#)  
 QUERY SYSCONTROL command [1201](#)  
 QUERY SYSOPER command [1202](#)  
 QUERY SYSTEM command [1203](#)  
 QUERY TAG command [1205](#)  
 QUERY TAPES command [1207](#)  
 QUERY TDISK command [1211](#)  
 QUERY TDISKCLR command [1212](#)  
 QUERY TDSK command [1211](#)  
 QUERY TERMINAL command [1213](#)  
 QUERY THROTTLE command [1217](#)  
 QUERY TIME command [1219](#)  
 QUERY TIMEZONES command [1191, 1222](#)  
 QUERY TOKEN command [1224](#)  
 QUERY TRACE command [1226](#)  
 QUERY TRACEFRAMES command [1230](#)  
 QUERY TRFILES command [1231](#)  
 QUERY TRSAVE command [1235](#)  
 QUERY TRSOURCE command [1239](#)  
 QUERY UCR command [1246](#)  
 QUERY UNDERSCORE command [1249](#)  
 QUERY UNRESOLVED command [1250](#)  
 QUERY UR command [1252](#)  
 QUERY USERID command [1261](#)  
 QUERY USERS command [1264](#)  
 QUERY VARIABLE command [1267](#)  
 QUERY VCONFIG command [1270](#)  
 QUERY VDISK command [1271](#)  
 QUERY VIRTUAL ALL command [1273](#)  
 QUERY VIRTUAL CHPID command [1275](#)  
 QUERY VIRTUAL CONSOLE command [1278](#)  
 QUERY VIRTUAL CPUS command [1282](#)  
 QUERY VIRTUAL CRYPTO command [1284](#)  
 QUERY VIRTUAL CTCA command [1287](#)  
 QUERY VIRTUAL DASD command [1288](#)  
 QUERY VIRTUAL DUPLEX command [1295](#)  
 QUERY VIRTUAL FCP command [1297](#)  
 QUERY VIRTUAL FLASHCOPY command [1300](#)  
 QUERY VIRTUAL GRAF command [1303](#)  
 QUERY VIRTUAL LINES command [1305](#)  
 QUERY VIRTUAL MSGDEVICES command [1307](#)  
 QUERY VIRTUAL MSGPROC [1308](#)  
 QUERY VIRTUAL NIC command [1310](#)  
 QUERY VIRTUAL OSA command [1316](#)  
 QUERY VIRTUAL PAV command [1319](#)  
 QUERY VIRTUAL PCIFUNCTION command [1322](#)  
 QUERY VIRTUAL PRINTER command [1324](#)  
 QUERY VIRTUAL PUNCH command [1328](#)  
 QUERY VIRTUAL READER command [1331](#)

QUERY VIRTUAL STORAGE command [1333](#)  
QUERY VIRTUAL SWITCHES command [1336](#)  
QUERY VIRTUAL TAPES command [1338](#)  
QUERY VIRTUAL UR command [1340](#)  
QUERY VMDUMP command [1341](#)  
QUERY VMLAN command [1344](#)  
QUERY VMRELOCATE command [1348](#)  
QUERY VMSG command [1349](#)  
QUERY VSWITCH command [1351](#)  
QUERY VTOD command [1370](#)  
QUERY WRKALLEG command [1372](#)

## R

rate, hot I/O  
    changing [1516](#)  
    querying [938](#)  
    setting [1516](#)  
RDR operand  
    DRAIN UR command [388](#)  
    LOCATE SPFBK command [628](#)  
reader  
    changing [1634](#)  
    changing options [1808](#)  
    clearing RDEV information [1636](#)  
    defining [1634](#)  
    displaying status [1252](#)  
    stopping spool operations [388](#)  
    unsupported  
        changing [1663](#)  
        clearing RDEV information [1636](#)  
        defining [1663](#)  
reader files  
    putting in order [2243](#)  
    spool  
        ending activity [70](#)  
READER operand  
    CLOSE command [70](#)  
    DEFINE (spooling device) command [209](#)  
    DRAIN UR command [388](#)  
    LOCATE SPFBK command [628](#)  
    SET RDEVICE command [1634](#), [1664](#)  
READER option  
    CLOSE command [73](#)  
READY command [1374](#)  
real  
    channel-to-channel adapters, displaying status [857](#)  
    communication lines, displaying status [976](#)  
    DASD  
        displaying status [867](#)  
    device block, locating [618](#)  
    device set, adding to [1627](#)  
    device set, changing [1627](#)  
    devices, displaying status [759](#), [1114](#)  
    display devices, displaying status [931](#)  
    dynamic switching device, displaying status [1194](#)  
    FCP devices, displaying status [911](#)  
    frame, locating frame table entry for [612](#)  
    magnetic tape drives, displaying status [1207](#)  
    printer, displaying status [1252](#)  
    punch, displaying status [1252](#)  
    reader, displaying status [1252](#)  
    unit record devices, stopping spool operations [388](#)  
real device

real device (*continued*)  
    adding [168](#)  
    attaching to host system [32](#)  
    changing [693](#)  
    control block, deleting [274](#)  
    creating [168](#)  
    defining [168](#)  
    deleting [265](#)  
    deleting control block [274](#)  
REAL operand  
    SET TIMER command [1737](#)  
real storage, host  
    displaying size [1186](#)  
    locating addresses of data in [630](#)  
    locking [654](#)  
    status, displaying [759](#)  
recall  
    guest operating system messages [1349](#)  
recording  
    accounting records  
        changing [1375](#)  
        collecting [2219](#)  
        displaying status [1130](#)  
    config records  
        changing [1375](#)  
        collecting [2219](#)  
        displaying status [1130](#)  
    EREP records  
        changing [1375](#)  
        collecting [2219](#)  
        displaying status [1130](#)  
    symptom records  
        changing [1375](#)  
        collecting [2219](#)  
        displaying status [1130](#)  
RECORDING command [1375](#)  
redefine  
    CP command [683](#)  
    device numbers [227](#)  
    DIAGNOSE code [697](#)  
    I/O configuration [122](#)  
    time zone [225](#)  
    virtual configuration [122](#)  
REDEFINE command [1378](#)  
refresh  
    logo configuration file information [1380](#)  
REFRESH command [1380](#)  
register  
    displaying contents [378](#)  
    dumping contents [433](#), [439](#)  
release  
    locked guest pages [2043](#)  
    minidisk accessed by CP [103](#)  
    spool files from system HOLD [511](#)  
    user from LOGON HOLD [510](#)  
relocation domain  
    DEFINE RELODOMAIN command [203](#)  
    QUERY RELODOMAIN command [1132](#)  
    QUERY VMRELOCATE command [1348](#)  
    SET VMRELOCATE command [1755](#)  
RELSpace command [1382](#)  
remove  
    closed spool files [740](#)  
    CP routines from the system execution space [117](#)

remove (*continued*)

- devices from a virtual machine [279](#)
- exit [700](#)
- guest LAN [279](#)
- image library files [743](#)
- logical devices from a virtual machine [279](#)
- logical devices from the host system [279](#)
- message repository files [745](#)
- NSS files [747](#)
- processors from a virtual machine [279](#)
- processors using DETACH CPU [284](#)
- real devices from a virtual machine [279](#)
- real devices from the host system [279](#)
- system data files [747](#)
- system trace files [749](#)
- UCR files [751](#)

REPEAT command [1384](#)

replace

- image library [2201](#)
- logo configuration file [1380](#)

reposition

- spool output [54](#), [508](#)

repository, message

- files, purging [745](#)

REQUEST command [1386](#)

RESERVE operand

- QUERY DASD command [869](#)

reserve release [1664](#)

RESERVE\_RELEASE operand

- SET RDEVICE command [1664](#)

reset

- account data [15](#)

RESET command [1387](#)

RESET operand

- SET OBSERVER command [1591](#)
- SET PRIVCLASS command [1616](#)
- SET SECUSER command [1688](#)

reset, power-on

- changing IOCDS file used at next [1525](#)

RESOURCE operand

- QUERY CONV command [810](#)

resources used by virtual machine, displaying [555](#)

RESPECT operand

- LOCATE (Storage) command [631](#)
- LOCATEVM command [652](#)

response examples, notation used in [6](#)

responses from command, suppressing [1798](#)

restart

- command execution [56](#)
- drained spool device [1867](#)
- message processor [1391](#)
- real DASD [1864](#)
- spool output [54](#)

RESTART MSGPROC [1391](#)

restore

- user minidisks [2149](#)

restricted User IDs [2263](#)

restriction

- DEFINE CNTLUNIT [162](#)
- DEFINE CU [162](#)
- DEFINE DEVICE [168](#)
- DEFINE IODEVICE [168](#)
- DELETE DEVICE [265](#)
- DELETE IODEVICE [265](#)

restriction (*continued*)

- for advanced function printers [1629](#)
- MODIFY CNTLUNIT [688](#)
- MODIFY CU [688](#)
- MODIFY DEVICE [693](#)
- MODIFY IODEVICE [693](#)
- QUERY CONFIGMODE [806](#)
- QUERY DYNAMIC\_I/O [895](#)
- SET RDEVICE command [1627](#)
- turning off hot I/O rate [1517](#)

retrieve

- guest operating system messages [1349](#)

retrieve buffer

- PF key
  - displaying [1143](#)
  - setting [1681](#)

RETRIEVE utility [2219](#)

retrieving Miscellaneous Data Records (MDRs) [1910](#)

rewind

- real device tapes [1393](#)

REWIND command [1393](#)

routing error records [2219](#)

run

- 370 applications in ESA/XA/XC machines [1787](#)

## S

SALIPL utility [2222](#)

sample

- data collection profile, changing [719](#)
- data collection profile, setting [719](#)
- monitoring [727](#)

SAPL (Stand-Alone Program Loader)

- installing copy [2222](#)

save

- DCSS in defined SDF [1394](#)
- NSS in defined SDF [1397](#)
- spool files on tape [1836](#)
- system data files on tape [1836](#)

SAVESEG command [1394](#)

SAVESYS command [1397](#)

saving data in system test files

- using TRSAVE [2011](#)

saving data on tape

- using TRSAVE [2011](#)

saving monitor records

- using MONWRITE [2212](#)

SCHEDULE [1401](#)

screen

- logo
  - adding to configuration file [1380](#)
  - replacing configuration file [1380](#)

SCREEN command [1404](#)

SCSIDISC utility [2228](#)

SDINST utility [2235](#)

search

- for data in virtual machine's storage [651](#)

SECLABEL operand

- START UR command [1871](#)

secondary user

- defining [1688](#)
- displaying [1153](#)
- setting [1688](#)



SECTABLE utility [2239](#)  
 security  
     function, journal [1533](#)  
     use of badge reader to log on to system [1646](#)  
 select  
     virtual processor [107](#)  
 selected lines of file, displaying [105](#)  
 send  
     cache fast write data [80](#)  
     DBCS data supported [1407](#)  
     messages [673](#), [676](#)  
     messages to all active users [674](#)  
     no-header messages [731](#)  
     special message [1800](#)  
     to a disconnected virtual machine [1407](#)  
 SEND command [1407](#)  
 sending commands to a guest operating system  
     using VINPUT [2071](#)  
 sending high-priority messages  
     using WARNING [2097](#)  
 SEP operand  
     START UR command [1869](#)  
 SEPARATOR operand  
     SET RDEVICE command [1651](#), [1669](#)  
 session  
     account number, changing [1416](#)  
     information, displaying current [1219](#)  
 set  
     DASD fast write [1485](#)  
     device recording mode [1671](#)  
     device-end interrupt pending [1374](#)  
     error message handling [1512](#)  
     error recording mode [1574](#)  
     event data collection profile [711](#)  
     hot I/O rate [1516](#)  
     informational message display [1519](#)  
     internal CP checking [1439](#)  
     interrupt rate, unsolicited [1516](#)  
     IOCDS file for next POR [1525](#)  
     logical line edit functions [1539](#)  
     maximum number of logged on users [1559](#)  
     number of PF key RETRIEVE buffers [1681](#)  
     password command entry [1598](#)  
     products, using SET PRODUCT [1619](#)  
     protected application environment [1434](#)  
     relocation domain [1755](#)  
     reserved pages (frames) [1674](#)  
     sample data collection profile [719](#)  
     secondary user [1688](#)  
     SET OBSERVER command [1591](#)  
     set real PCI function [1600](#)  
     shutdown signal disablement [1700](#)  
     spool file class for impact printers [1649](#)  
     system abend dump location [1493](#)  
     system storage for virtual disks [1749](#)  
     tab function values [1609](#)  
     unsolicited interrupt rate [1516](#)  
     user access to system resources [1625](#)  
     user frame owned list reordering [1672](#)  
     user ID of primary system operator [1731](#)  
     virtual processor ID [1474](#)  
 SET 370ACCOM command [1787](#)  
 SET ABEND command [1413](#)  
 SET ACCOUNT command [1416](#)  
 SET ADJUNCTS command [1417](#)  
 SET AGELIST command [1418](#)  
 SET AUTOPOLL command [1421](#), [1736](#)  
 SET CACHE command [1422](#)  
 SET CACHEFW command [1427](#)  
 SET CFLINK [1430](#)  
 SET CMDLIMIT command [1432](#)  
 SET command [1411](#)  
 SET CONCEAL command [1434](#)  
 SET CONFIGMODE command [1436](#)  
 SET CPCHECKING command [1439](#)  
 SET CPCONIO command [1440](#)  
 SET CPLANGUAGE command [1441](#)  
 SET CPPROTECT command [1443](#)  
 SET CPTRACE command [1446](#)  
 SET CPUAFFINITY command [1472](#)  
 SET CPUID command [1474](#)  
 SET CPUPOOL command superseded by SET RESPOOL  
 command [1679](#)  
 SET CRYPTO command [1476](#)  
 SET CU command [1478](#)  
 SET D8ONECMD command [1482](#)  
 SET DASD command [1484](#)  
 SET DASDFW command [1485](#)  
 SET DATEFORMAT command [1488](#)  
 SET DEVICES command [1490](#)  
 SET DIALDROP command [1492](#)  
 SET DUMP command [1493](#)  
 SET DUMPDEV command [1496](#)  
 SET DYNAMIC\_I/O command [1502](#)  
 SET EDEVICE command [1504](#)  
 SET EMSG command [1512](#)  
 SET ENCRYPT command [1513](#)  
 SET HOTIO command [1516](#)  
 SET IMMSG command [1519](#)  
 SET IO\_OPT command [1521](#)  
 SET IOASSIST command [1523](#)  
 SET IOCDS\_ACTIVE command [1525](#)  
 SET IPLPARMS command [1530](#)  
 SET JOURNAL command [1533](#)  
 SET KEYALIAS command [1534](#)  
 SET LAN command [1536](#)  
 SET LINEDIT command [1539](#)  
 SET LKFAC command [1540](#)  
 SET LKFACR command [1542](#)  
 SET LOADDEV command [1544](#)  
 SET LOGMSG command [1551](#)  
 SET LSYSTEM command [1553](#)  
 SET MACHINE command [1555](#)  
 SET MAXLDEV [1558](#)  
 SET MAXUSERS command [1559](#)  
 SET MDCACHE command [1562](#)  
 SET MEMASSIST command [1568](#)  
 SET MIH command [1570](#)  
 SET MITIME command [1571](#)  
 SET MODE command [1574](#)  
 SET MONDATA command [1575](#)  
 SET MSG command [1576](#)  
 SET MSGFACIL command [1577](#)  
 SET MULTITHREAD command [1578](#)  
 SET NEW\_DEVICES command [1581](#)  
 SET NOPDATA command [1587](#)  
 SET NVS command [1588](#)  
 SET operand

SET operand (*continued*)

- LOCATE CMDBK command [603](#)
- QUERY CPCMDs command [818](#)
- SET PAGEX command [1594](#)
- SET PAGING command [1596](#)
- SET PASSWORD command [1598](#)
- SET PCIFUNCTION command [1600](#)
- SET PFnn command [1602](#)
- SET PFnn COPY command [1605](#)
- SET PFnn RETRIEVE command [1607](#)
- SET PFnn TAB command [1609](#)
- SET PORT GROUP command [1610](#)
- SET PRIVCLASS command [1616](#)
- SET PRODUCT command [1619](#)
- SET PROMPT command [1621](#)
- SET PSWTRANS command [1622](#)
- SET QIOASSIST command [1623](#)
- SET QUICKDSP command [1625](#)
- SET RDEVICE command
  - 3215 terminal [1661](#)
  - 3800 printers [1667](#)
  - advanced function printers [1629](#)
  - card punches [1631](#)
  - card readers [1634](#)
  - channel-to-channel adapter [1655](#)
  - communication controllers [1638](#)
  - DASD [1641](#)
  - FCP device [1655](#)
  - general description [1627](#)
  - graphic display devices [1645](#)
  - HiperSockets device [1655](#)
  - impact printers [1648](#)
  - integrated communication adapters [1653](#)
  - line adapters [1638](#)
  - OSA device [1655](#)
  - special devices [1655](#)
  - tape units [1658](#)
  - teleprocessing communications adapters [1653](#)
  - terminals [1661](#)
  - unsupported devices [1663](#)
- SET RECORD command [1134](#), [1671](#), [1672](#)
- SET RESERVED command [1674](#)
- SET RESPOOL command [1679](#)
- SET RETRIEVE command [1681](#)
- SET RUN command [1683](#)
- SET SCMEASURE command [1685](#)
- SET SECUSER command [1688](#)
- SET SHARE command [1691](#)
- SET SHARED command [1696](#)
- SET SHUTDOWNTIME command [1698](#)
- SET SHUTSIGNAL command [1700](#)
- SET SIGNAL command [1701](#)
- SET SMSG command [1703](#)
- SET SRM command [1705](#)
- SET SSI command [1714](#)
- SET STGEXEMPT command [1719](#)
- SET STGLIMIT command [1721](#)
- SET STORAGE command [1722](#)
- SET SVC76 command [1729](#)
- SET SYSCONTROL command [1730](#)
- SET SYSOPER command [1731](#)
- SET THROTTLE command [1734](#)
- SET TIMER command [1737](#)
- SET TIMEZONE command [1738](#)
- SET TOKEN command [1740](#)
- SET TRACEFRAMES command [1742](#)
- SET UNDERSCORE command [1744](#)
- SET VARIABLE command [1745](#)
- SET VCONFIG command [1747](#)
- SET VDISK command [1749](#)
- SET VMCONIO command [1751](#)
- SET VMLAN command [1752](#)
- SET VMRELOCATE command [1755](#)
- SET VSWITCH command [1757](#)
- SET VTOD command [1781](#)
- SET WNG command [1784](#)
- SET WRKALLEG command [1785](#)
- SET\_ADDRESS operand
  - SET RDEVICE command [1638](#)
- setting virtual machine activity pause
  - using SLEEP [1799](#)
- setting virtual switch [1757](#)
- settings
  - host encryption [904](#)
- SETUP operand
  - START UR command [1869](#)
- SFCOUNTR utility [2241](#)
- share
  - full-pack minidisks [1696](#)
- shared DASD
  - changing [1641](#)
  - clearing RDEV information [1636](#)
  - defining [1641](#)
- shared data
  - reserve/release [1664](#)
- SHARED operand
  - DEFINE CPOWNER command [150](#)
  - SET RDEVICE command [1641](#)
- shared system name table, locating [623](#)
- show
  - access modes to search for system logmsg [988](#)
  - active partition [991](#)
  - address
    - control information [602](#)
    - CP command table entry block [603](#), [817](#)
  - all
    - partitions [991](#)
    - paths installed to a device [1064](#)
  - buffers, PF key retrieve [1143](#)
  - channel-to-channel adapters [857](#)
  - communication lines [976](#)
  - CP-owned volume list [834](#)
  - current time zone definitions [1222](#)
  - cylinder allocation [773](#)
  - D8ONECMD settings [865](#)
  - DASD [867](#)
  - default reorder setting [1134](#)
  - display devices [931](#)
  - dynamic switching devices [1194](#)
  - FCP [911](#)
  - HCD status [934](#)
  - hot I/O rate [938](#)
  - HYPERSWAP status [941](#)
  - interrupt rate, unsolicited [938](#)
  - IPL parameters, system [954](#)
  - key retrieve buffers, PF [1143](#)
  - location
    - control information [602](#)

show (*continued*)

- location (*continued*)
  - CP command table entry block [603](#), [817](#)
  - CP module [832](#)
  - CP nucleus [832](#)
  - parm disk [832](#)
- log message [988](#)
- minidisk device characteristics [1004](#)
- Open Systems Adapters [1055](#)
- page allocation [773](#)
- partitions [991](#)
- PF key
  - retrieve buffers [1143](#)
- privilege class [1090](#)
- relocation domain [203](#), [1132](#), [1348](#)
- retrieve buffers, PF key [1143](#)
- server time protocol [1191](#)
- space allocated for internal trace tables [1230](#)
- spool file information [1104](#)
- status
  - all devices [759](#)
  - dynamic I/O support [895](#)
  - host real storage [759](#)
  - installed device paths [1064](#)
  - journaling, security [963](#)
  - logical devices [973](#)
  - measuring, subchannel [1149](#)
  - Parallel Access Volume DASD [867](#), [1068](#)
  - print server virtual machine [1089](#)
  - real channel-to-channel adapters [857](#)
  - real communication lines [976](#)
  - real DASD [867](#)
  - real devices [1114](#)
  - real display devices [931](#)
  - real dynamic switching devices [1194](#)
  - real magnetic tape drives [1207](#)
  - security journaling [963](#)
  - subchannel measuring [1149](#)
  - unit record devices [1252](#)
- storage
  - available for dynamic I/O changes [895](#)
- system operator [1202](#)
- system scheduler/dispatcher [531](#)
- unsolicited interrupt rate [938](#)
- user ID of system operator [1202](#)
- SHUTDOWN command [1789](#)
- shutdown signal disablement [1700](#)
- SIGNAL command [1796](#)
- SILENTLY command [1798](#)
- simulate
  - external interrupt pending [469](#)
  - I/O interrupt [1570](#)
  - IPL in virtual machine [566](#)
- SINGLE operand [191](#)
- size
  - changing storage [213](#)
  - defining storage [213](#)
- SIZE operand
  - DEFINE (spooling device) command [210](#)
- SLEEP command [1799](#)
- SLOT operand
  - DEFINE CPOWNERD command [150](#)
- SMSG command [1800](#)
- SNA control block, locating [625](#)
- SNA operand
  - DISABLE (device) command [311](#)
- SNAPDUMP command [1802](#)
- space
  - allocating
    - for internal trace tables [1742](#)
    - for new spool records, preventing [386](#)
  - querying
    - for internal trace tables [1230](#)
- SPACE command [1804](#)
- SPACE operand
  - LOCK command [654](#)
  - UNLOCK command [2043](#)
- special devices
  - changing [1655](#)
  - clearing RDEV information [1636](#)
  - defining [1655](#)
- SPECIAL operand
  - SET HOTIO command [1517](#)
- specify
  - DASD fast write [1485](#)
  - device recording mode [1671](#)
  - device-end interrupt pending [1374](#)
  - error message handling [1512](#)
  - error recording mode [1574](#)
  - event data collection profile [711](#)
  - FCB image [598](#)
  - hot I/O rate [1516](#)
  - informational message display [1519](#)
  - internal CP checking [1439](#)
  - interrupt rate, unsolicited [1516](#)
  - IOCDS file for next POR [1525](#)
  - log message text [1551](#)
  - logical line edit functions [1539](#)
  - maximum number of logged on users [1559](#)
  - number of PF key RETRIEVE buffers [1681](#)
  - password command entry [1598](#)
  - products, using SET PRODUCT [1619](#)
  - protected application environment [1434](#)
  - relocation domain [1755](#)
  - reserved pages (frames) [1674](#)
  - sample data collection profile [719](#)
  - secondary user [1688](#)
  - SET OBSERVER command [1591](#)
  - shutdown signal disablement [1700](#)
  - spool file class for impact printers [1649](#)
  - system abend dump location [1493](#)
  - system storage for virtual disks [1749](#)
  - tab function values [1609](#)
  - unsolicited interrupt rate [1516](#)
  - user access to system resources [1625](#)
  - user frame owned list reordering [1672](#)
  - user ID of primary system operator [1731](#)
  - virtual processor ID [1474](#)
- SPOL operand
  - START (disk) command [1864](#)
- SPOL option
  - DRAIN (disk) command [386](#)
- spool
  - changing options [1808](#)
  - consoles
    - ending activity [70](#)
  - operations, stopping on real unit record devices [388](#)

- spool (*continued*)
  - records
    - preventing CP from allocating space for [386](#)
- SPOOL command [1808](#)
- spool device
  - restarting drained [1867](#)
- spool file
  - block, locating [627](#)
  - classes
    - assigning [70](#)
    - specifying for impact printers [1649](#)
  - commands
    - CLOSE [70](#)
    - DRAIN (disk) [386](#)
    - DRAIN UR [388](#)
  - displaying [1104](#)
  - displaying count [917](#)
  - displaying limits [997](#)
  - displaying tag information [1205](#)
  - identification entry, locating [610](#)
  - loading from tape [1846](#)
  - order, changing queued [735](#)
  - printer
    - ending activity [70](#)
  - punch
    - ending activity [70](#)
  - purging closed [740](#)
  - reader
    - ending activity [70](#)
  - saving on tape [1836](#)
- spool files
  - putting in order [2243](#)
- SPOOL operand
  - QUERY ALLOC command [774](#)
  - START (disk) command [1864](#)
- SPOOL option
  - DRAIN (disk) command [386](#)
- spooling
  - device [209](#)
  - options [1809](#)
- SPORDER utility [2243](#)
- SPXTAPE CANCEL command [1834](#)
- SPXTAPE command [1818](#)
- SPXTAPE DUMP command [1836](#)
- SPXTAPE END command [1844](#)
- SPXTAPE LOAD command [1846](#)
- SPXTAPE SCAN command [1857](#)
- SRM (system resource management)
  - displaying system-wide parameters [1173](#)
- start
  - command execution [56](#)
  - event monitoring [727](#)
  - journal security function [1533](#)
  - monitoring of sample data collection [719](#)
  - real DASD [1864](#)
  - sample monitoring [727](#)
- START command
  - for DASD [1864](#)
  - for unit record devices [1867](#)
- START DASD command [1864](#)
- START VOLUME command [1864](#)
- STATE operand
  - QUERY CONV command [810](#)
- STATISTICS operand
  - CPLISTFILE command [100](#)
- status
  - displaying channel path [791](#), [796](#), [798](#), [1275](#)
  - displaying virtual DASD [1288](#)
  - displaying virtual disk in storage [1288](#)
  - of CP-accessed disk, displaying [824](#)
  - of host real storage, displaying [759](#)
  - of Parallel Access Volume DASD, querying
    - QUERY DASD DETAILS command [867](#)
    - QUERY PAV command [1068](#)
  - of real DASD, querying [867](#)
  - of real devices, displaying [759](#)
  - setting enable or disable status of products [1619](#)
  - subchannel measuring [1149](#)
- stop
  - active channel program [520](#)
  - all virtual processors (soft) [1878](#)
  - ATTACH commands [96](#)
  - channel program, active [520](#)
  - console sharing [1688](#)
  - event monitoring [727](#)
  - journal security function [1533](#)
  - LOCATE commands [96](#)
  - LOCATEVM commands [96](#)
  - monitoring of sample data collection [719](#)
  - new operations on real DASD [386](#)
  - processing tape files [1844](#)
  - program, active channel [520](#)
  - sample monitoring [727](#)
  - selected real processors (hard) [1878](#)
- spool
  - activity on virtual spool devices [70](#)
  - file [498](#)
  - operations on real unit record devices [388](#)
  - system functions [1789](#)
  - tape procedures [1834](#)
  - use of files on minidisk accessed by CP [103](#)
  - VARY commands [96](#)
  - virtual machine session [659](#)
- STOP command [1878](#)
- stopping MONWRITE processing
  - using MONWSTOP [2216](#)
- storage
  - available amount
    - displaying for channel paths [895](#)
    - displaying for control units [895](#)
    - displaying for devices [895](#)
    - displaying for dynamic I/O changes [895](#)
    - displaying for subchannels [895](#)
  - changing size [213](#)
  - defining size [213](#)
  - displaying contents of first-level (host) storage [363](#)
  - displaying size of real [1186](#)
  - displaying status [759](#)
  - dumping contents of first-level (host) storage [424](#)
  - host, displaying addresses of data in [630](#)
  - HSA
    - displaying amount available [895](#)
  - limit detection, controlling free [1719](#), [1721](#)
  - locking [654](#)
  - making nonvolatile available [1588](#)
  - search for data in virtual machine's [651](#)

storage (*continued*)  
 storing data into first-level (host) storage [1898](#)  
 storing data into guest (ESA/390) [1886](#)  
 storing data into guest (ESA/XC, z/XC) [1881](#)  
 storing data into guest (z/Architecture) [1892](#)  
 virtual disks, controlling system [1749](#)

store  
 data into first-level (host) storage [1898](#)  
 data into guest storage (ESA/390) [1886](#)  
 data into guest storage (ESA/XC, z/XC) [1881](#)  
 data into guest storage (z/Architecture) [1892](#)  
 data into low storage [1908](#)  
 data into registers [1905](#)  
 PSW data [1901](#)  
 PSWA data [1903](#)

STORE (Guest Storage - ESA/390) command [1886](#)  
 STORE (Guest Storage - ESA/XC, z/XC) command [1881](#)  
 STORE (Guest Storage - z/Architecture) command [1892](#)  
 STORE (Host Storage) command [1898](#)  
 STORE PSW/CAW/CSW STORAGE command [1901](#), [1903](#)  
 STORE REGISTERS command [1905](#)  
 STORE STATUS command [1908](#)

SUBCMD operand  
 LOCATE CMDDBK command [603](#)  
 QUERY CPCMDS command [817](#)

SUBSYSTEM operand  
 DESTAGE command [277](#)  
 SET CACHE command [1422](#)  
 SET CACHEFW command [1427](#)  
 SET NVS command [1588](#)

subsystem, channel  
 disabling [2067](#)  
 displaying storage available for adding [895](#)  
 enabling [2067](#)  
 TRACE [1954](#)  
 tracing I/O instructions for [1452](#)

suppress  
 trace events [2003](#)

suppressing command responses [1798](#)

SWITCH operand  
 SET HOTIO command [1517](#)  
 SET RDEVICE command [1664](#)

switching device, dynamic  
 displaying status [1194](#)  
 unsupported  
 changing [1663](#)  
 clearing RDEV information [1636](#)  
 defining [1663](#)

SXSTE, locating [634](#)

symbol  
 address, locating system external [636](#)

SYMBOL operand  
 LOCK command [655](#)  
 UNLOCK command [2044](#)

SYMPTOM operand  
 RECORDING command [1375](#)  
 RETRIEVE utility [2219](#)

symptom records, recording  
 changing [1375](#)  
 collecting [2219](#)  
 displaying status [1130](#)

SYNCHRONOUS operand  
 CPACCESS command [90](#)

SYNCHRONOUS operand (*continued*)  
 CPRELEASE command [103](#)

SYNCHRONOUS option  
 CLOSE command [73](#)

SYNCDMS command [1910](#)

syntax  
 checking configuration file [2143](#)  
 syntax diagrams, how to read [4](#)

SYSASCII operand, ATTACH [32](#)

SYSCTL operand  
 ATTACH command [35](#)

system  
 definition  
 of set of real devices [1627](#)  
 external symbol address, locating [636](#)  
 frame table entry, locating [612](#)  
 logical device block, locating [616](#)  
 parameters, changing [1705](#)  
 real device block, locating [618](#)  
 resources, assigning user access to [1625](#)  
 shared name table, locating [623](#)  
 SNA control block, locating [625](#)  
 storage for virtual disks, controlling [1749](#)  
 VTAM service machine block, locating [647](#)

SYSTEM command [1911](#)

system data file  
 CHANGE [67](#)  
 defining for named saved system [253](#)  
 defining for saved segment [246](#)  
 displaying count of [1152](#)  
 displaying information [1041](#)  
 IPL [571](#)  
 loading from tape [1846](#)  
 PURGE NSS [747](#)  
 purging [743](#)  
 QUERY IMG [946](#)  
 QUERY NLS [1039](#)  
 QUERY TRFILES [1231](#), [1233](#)  
 SAVESEG [1394](#), [1395](#)  
 SAVESYS [1397](#)  
 saving NSS into [1397](#)  
 saving on tape [1836](#)  
 TRANSFER [2010](#)  
 TRSOURCE [2019](#)

system execution space  
 address range, displaying [1199](#)  
 loading CP routines into [111](#)  
 page status, displaying [1196](#)  
 unloading CP routines from [117](#)

system function  
 ending [1789](#)

system loader  
 controlling [2271](#)

system log message  
 displaying [988](#)

SYSTEM operand  
 ACNT command [15](#)  
 ATTACH command [34](#)  
 LOCATE SPFBK command [627](#)  
 QUERY ALL command [760](#)  
 QUERY DASD command [869](#)  
 QUERY LOGMSG command [988](#)  
 QUERY MAXSPOOL [997](#)  
 QUERY MDCACHE command [1000](#)

SYSTEM operand (*continued*)  
  QUERY TAPES command [1208](#)  
  QUERY UR command [1252](#)  
  SET MDCACHE command [1562](#)  
system operator  
  changing user ID of [1731](#)  
  displaying user ID of [1202](#)  
System Software Inventory files  
  updating [2211](#)

## T

table  
  effects of file status, device options, and CLOSE  
  command options on reader files [74](#)  
  shared system name, displaying [623](#)  
table entry block, command  
  defining new CP [143](#)  
  displaying address for CP command [603](#), [817](#)  
  locating for CP command [603](#), [817](#)  
table, trace  
  allocating space for  
    internal [1742](#)  
  displaying space allocated for [1230](#)  
TAG command [1914](#)  
tape  
  drives  
    magnetic, displaying status [1207](#)  
  dumping files to [1836](#)  
  file names, scanning [1857](#)  
  files, stop processing of [1844](#)  
  loading files from [1846](#)  
  spool file processing [1818](#)  
  stopping procedures [1834](#)  
  units  
    changing [1658](#)  
    clearing RDEV information [1636](#)  
    defining [1658](#)  
    unsupported, changing [1663](#)  
    unsupported, clearing RDEV information [1636](#)  
    unsupported, defining [1663](#)  
TAPE operand  
  SET HOTIO command [1517](#)  
  SET RDEVICE command [1664](#)  
TDISK command [223](#)  
TDISK operand  
  QUERY ALLOC command [774](#)  
  QUERY SYSTEM command [1203](#)  
  START (disk) command [1865](#)  
TDISK option  
  DRAIN (disk) command [387](#)  
TDSK operand  
  QUERY SYSTEM command [1203](#)  
  START (disk) command [1865](#)  
TDSK option  
  DRAIN (disk) command [387](#)  
TELE2 operand  
  DEFINE LINE command [186](#)  
TELE2\_ADAPTER operand  
  SET RDEVICE command [1638](#)  
teleprocessing integrated adapter  
  changing [1653](#)  
  clearing RDEV information [1636](#)  
  defining [1653](#)

TEMPDISK operand  
  START (disk) command [1865](#)  
TEMPDISK option  
  DRAIN (disk) command [387](#)  
temporary disk (TDISK)  
  adding [223](#)  
  defining [223](#)  
  displaying space [1211](#)  
  preventing CP from allocating [386](#)  
terminal  
  HOLDING status [1932](#)  
  MORE status [1937](#)  
  options, displaying [1213](#)  
  unsupported  
    changing [1663](#)  
    clearing RDEV information [1636](#)  
    defining [1663](#)  
TERMINAL APL  
  command [1919](#)  
  DBCS supported [1919](#)  
TERMINAL ASCIITBL command [1920](#)  
TERMINAL ATTN command [1921](#)  
TERMINAL AUTOOCR command [1922](#)  
TERMINAL BREAKIN command [1923](#)  
TERMINAL BRKKEY command [1924](#)  
TERMINAL CHARDEL command [1926](#)  
TERMINAL CNTL command [1927](#)  
TERMINAL command [1917](#)  
TERMINAL CONMODE command [1928](#), [1941](#)  
TERMINAL ESCAPE command [1930](#)  
TERMINAL HILIGHT command [1931](#)  
TERMINAL HOLD command [1932](#)  
TERMINAL LINEDEL command [1933](#)  
TERMINAL LINEND command [1934](#)  
TERMINAL LINESIZE command [1935](#)  
TERMINAL MODE command [1936](#)  
TERMINAL MORE command [1937](#)  
TERMINAL operand  
  SET HOTIO command [1517](#)  
  SET RDEVICE command [1664](#)  
TERMINAL PRECEDENCE  
  command [1938](#)  
TERMINAL PRECEDENCE command [1938](#)  
TERMINAL PROMPT command [1939](#)  
TERMINAL SCROLL command [1940](#)  
TERMINAL TABCHAR command [1942](#)  
TERMINAL TEXT  
  command [1943](#)  
  DBCS supported [1943](#)  
TERMINAL TYPE command [1945](#)  
terminals  
  changing [1661](#)  
  defining [1661](#)  
terminate  
  active channel program [520](#)  
  all virtual processors (soft) [1878](#)  
  ATTACH commands [96](#)  
  channel program, active [520](#)  
  console sharing [1688](#)  
  event monitoring [727](#)  
  journal security function [1533](#)  
  LOCATE commands [96](#)  
  LOCATEVM commands [96](#)  
  monitoring of sample data collection [719](#)

terminate (*continued*)  
   new operations on real DASD [386](#)  
   processing tape files [1844](#)  
   program, active channel [520](#)  
   sample monitoring [727](#)  
   selected real processors (hard) [1878](#)  
   spool  
     activity on virtual spool devices [70](#)  
     file [498](#)  
     operations on real unit record devices [388](#)  
   system functions [1789](#)  
   tape procedures [1834](#)  
   use of files on minidisk accessed by CP [103](#)  
   VARY commands [96](#)  
   virtual machine session [659](#)  
 TEXT operand  
   SET EMSG command [1512](#)  
 thread information, displaying [1026](#), [1578](#)  
 time zone  
   changing [225](#), [1738](#)  
   defining [225](#)  
   displaying [1191](#), [1222](#)  
   redefining [225](#)  
   setting [1738](#)  
 timer  
   controlling interval [1737](#)  
 TO operand  
   QUERY PATHS command [1064](#)  
 TOKEN, set [1740](#)  
 trace  
   command, general description [1946](#)  
   data [2018](#)  
   displaying settings [1226](#)  
   real machine events [1446](#)  
   VM events [739](#)  
 TRACE  
   command responses [1954](#)  
 TRACE AIF [1963](#)  
 TRACE ALL command [1964](#)  
 TRACE APPEND command [1965](#)  
 TRACE AR command [1966](#)  
 TRACE BRANCH command [1968](#)  
 TRACE CLEAR command [1971](#)  
 TRACE COUNT command [1972](#)  
 TRACE DELETE command [1973](#)  
 TRACE END command [1974](#)  
 TRACE EXTERNAL/PROGRAM command [1975](#)  
 TRACE G command [1976](#), [1978](#)  
 TRACE GOTO/CALL/RETURN command [1980](#)  
 TRACE I/O command [1983](#)  
 TRACE INSTRUCTION command [1982](#)  
 TRACE MCH command [1985](#)  
 TRACE mnemonic1 command [1986](#)  
 TRACE mnemonic2 command [1988](#)  
 TRACE options [1949](#)  
 TRACE RIO/RSCH/SIO/SIOF/SSCH command [1989](#)  
 trace set [1946](#)  
 TRACE STORE command [1991](#), [1995](#)  
 TRACE SVC/DIAGNOSE/MC command [1998](#)  
 trace table  
   allocating space for  
     internal [1742](#)  
   displaying space allocated for [1230](#)  
 TRACE TABLE (*continued*)  
   command [1999](#)  
 TRACE TPI command [2001](#)  
 trace trap  
   usage information [1946](#)  
 TRACE TRAP command [2002](#)  
 TRACE TXSUSPEND/NOTXSUSPEND command [2003](#)  
 TRACERED utility  
   exit descriptions [2252](#)  
   formatting exit [2253](#)  
   output [2249](#)  
   output routine interface [2254](#)  
 tracing access register changes  
   using TRACE AR [1966](#)  
 tracing alterations to virtual storage  
   using TRACE STORE [1991](#), [1995](#)  
 tracing assembler instruction execution  
   using TRACE mnemonic1 [1986](#)  
 tracing DIAGNOSE instructions  
   using TRACE SVC/DIAGNOSE/MC [1998](#)  
 tracing external interrupts  
   using TRACE EXTERNAL [1975](#)  
 tracing G alterations  
   using TRACE G [1976](#)  
 tracing GG alterations  
   using TRACE GG [1978](#)  
 tracing I/O instructions  
   using TRACE I/O [1983](#)  
   using TRACE mnemonic2 [1988](#)  
 tracing I/O interrupts  
   using TRACE I/O  
     [1983](#)  
 tracing instructions  
   using TRACE ALL [1964](#)  
   using TRACE INSTRUCTION [1982](#)  
 tracing interrupts  
   using TRACE ALL [1964](#)  
 tracing monitor calls  
   using TRACE SVC/DIAGNOSE/MC [1998](#)  
 tracing program interrupts  
   using TRACE PROGRAM [1975](#)  
 tracing RIO instruction  
   using TRACE RIO [1989](#)  
 tracing RSCH instruction  
   using TRACE RSCH [1989](#)  
 tracing SIO instruction  
   using TRACE SIO [1989](#)  
 tracing SIOF instruction  
   using TRACE SIOF [1989](#)  
 tracing SSCH instruction  
   using TRACE SSCH [1989](#)  
 tracing successful branching instructions  
   using TRACE BRANCH command [1968](#)  
 tracing supervisor calls  
   using TRACE SVC/DIAGNOSE/MC [1998](#)  
 tracing TPI instructions  
   using TRACE TPI [2001](#)  
 trademarks [2284](#)  
 transfer  
   control of a dedicated tape drive [513](#), [518](#)  
   control of a real tape drive [515](#)  
   files to virtual devices [2005](#)  
 TRANSFER command [2005](#)  
 TRSAVE command [2011](#)

TRSOURCE command [2018](#)  
 TRSOURCE DISABLE command [2022](#)  
 TRSOURCE DISPLAY command [2023](#)  
 TRSOURCE DROP command [2024](#)  
 TRSOURCE ENABLE command [2025](#)  
 TTY operand  
     TERMINAL TYPE command [1945](#)  
 TYPE operand  
     DEFINE CNTLUNIT command [162](#)  
     DEFINE CU command [162](#)  
     QUERY CONV command [810](#)  
     QUERY OSA command [1056](#)  
     SET RDEVICE command  
         3800 printers [1667](#)  
         advanced function printers [1629](#)  
         card punches [1631](#)  
         card readers [1634](#)  
         communication controllers [1638](#)  
         DASD [1641](#)  
         graphic display devices [1645](#)  
         impact printers [1649](#)  
         line adapters [1638](#)  
         special devices [1655](#)  
         terminals [1661](#)  
         unsupported devices [1664](#)

## U

UNCOUPLE command [2040](#)  
 underscore separator [1249](#), [1744](#)  
 UNDIAL command [2042](#)  
 unit address  
     adding to control unit [688](#)  
     deleting from control unit [688](#)  
 unit record  
     devices  
         displaying status [1252](#)  
         draining [388](#)  
 UNIT\_RECORD operand  
     SET HOTIO command [1517](#)  
 UNIVERSAL\_CHARSET operand  
     SET RDEVICE command [1651](#)  
 unload  
     CP routines from the system execution space [117](#)  
 unlock [2043](#)  
 UNLOCK operand  
     SET D8ONECMD command [1483](#)  
     SET PRIVCLASS command [1616](#)  
 unsolicited interrupt  
     changing [1516](#)  
     querying [938](#)  
     setting [1516](#)  
 unsupported devices  
     changing [1663](#)  
     clearing RDEV information [1636](#)  
     defining [1663](#)  
 UNSUPPORTED operand  
     SET RDEVICE command [1664](#)  
 UR operand  
     DRAIN UR command [388](#)  
     SET HOTIO command [1517](#)  
 user class restructure [1246](#)  
 user ID  
     system operator

user ID (*continued*)  
     system operator (*continued*)  
         changing [1731](#)  
         displaying [1202](#)  
 User IDs (restricted) [2263](#)  
 user minidisks  
     displaying device characteristics [1004](#)  
 USER operand  
     DEFINE CTCA command [159](#)  
     START UR command [1868](#)  
 USERID operand  
     LOCK command [654](#)  
     QUERY CONV command [810](#)  
     QUERY GATEWAY command [929](#)  
     QUERY LOGMSG command [989](#)  
     QUERY MDISK command [1004](#)  
     QUERY OBSERVER command [1052](#)  
     QUERY RESOURCE command [1138](#)  
     SET OBSERVER command [1591](#)  
     UNLOCK command [2043](#)  
 USERS operand  
     QUERY NSS command [1042](#)  
 utilities  
     CCDUMP [2114](#)  
     CCLOAD [2116](#)  
     CPEREPXA [2118](#)  
     CPFMTXA [2124](#)  
     CPSYNTAX [2143](#)  
     CVTOVRID [2147](#)  
     DDR [2149](#)  
     DIRECTXA [2168](#)  
     DISKMAP [2178](#)  
     DUMPLD2 [2184](#)  
     DUMpload [2180](#)  
     GENIMAGE [2192](#)  
     HCPLDR [2195](#)  
     IMAGELIB [2201](#)  
     IOCP [2203](#)  
     IOEXPLOR [2205](#)  
     IPWIZARD [2210](#)  
     MIGR51D [2211](#)  
     MONWRITE [2212](#)  
     MONWSTOP [2216](#)  
     MOVE2SFS [2217](#)  
     RETRIEVE [2219](#)  
     SALIPL [2222](#)  
     SCSIDISC [2228](#)  
     SDINST [2235](#)  
     SECTABLE [2239](#)  
     SFCOUNTR [2241](#)  
     SPORDER [2243](#)  
     TRACERED [2244](#)  
     UTILITY [2255](#)  
 UTILITY utility [2255](#)

## V

VAFP (virtual advanced function printer)  
     device characteristics [209](#)  
 VARY (real device) command [2063](#)  
 VARY CHPID command [2046](#)  
 VARY command  
     stopping [96](#)  
 VARY CORE command [2048](#)



VARY CRYPTO command [2050](#)  
 VARY ONLINE/OFFLINE PCIFUNCTION command [2059](#)  
 VARY PATH command [2053](#)  
 VARY PROCESSOR command [2061](#)  
 VARY SUBCHANNEL command [2067](#)  
 VDELETE command [2069](#)  
 verifying CSE/SSI system table matches CP table [2111](#)  
 version  
   creating new for existing CP [122](#), [143](#)  
 VINPUT command [2071](#)  
 virtual device  
   adding [227](#)  
   defining [227](#)  
   displaying [639](#)  
 virtual disk in storage  
   controlling system storage for [1749](#)  
   creating [230](#)  
   displaying status [1288](#)  
 virtual machine  
   configuration, changing [122](#)  
   controlling print server, displaying status [1089](#)  
   definition  
     locating block [642](#)  
   resources, displaying [555](#)  
   storage  
     searching for data in [651](#)  
 VIRTUAL operand  
   LOCATE CMDDBK command [603](#)  
   QUERY CPCMDS command [817](#)  
 virtual reader files  
   putting in order [2243](#)  
 virtual storage, user  
   locking [654](#)  
 virtual switch segment, detaching [307](#)  
 virtual switch segment, querying [1351](#)  
 virtual switch, defining [232](#)  
 virtual switch, setting [1757](#)  
 VMDUMP command [2073](#)  
 VMRELOCATE  
   VMRELOCATE command [2084](#)  
 VOLID operand  
   SALIPL command [2224](#)  
 volume list  
   changing [150](#)  
   CP-owned, displaying [834](#)  
   defining [150](#)  
 VOLUME operand  
   DRAIN (disk) command [386](#)  
   QUERY CPOWNED command [834](#)  
   START (disk) command [1864](#)  
 VSM operand  
   LOCATE SNABK command [625](#)  
 VTAM service machine (VSM)  
   locating block [647](#)

## W

warm start  
   checking system for [1789](#)  
 warning  
   calling entry point after loading CP routines [112](#)  
   changing status area fields on running system [1380](#)  
   changing system's local time zone [1738](#)  
   changing time zone [1738](#)

warning (*continued*)  
   CONTROL operand of CPXLOAD [112](#)  
   DBCS supported [2097](#)  
   FORCEOFF operand of SET DASDFW command [1485](#)  
   increasing size of internal trace tables [1742](#)  
   linking to another disk [579](#)  
   message, controlling display of [1784](#)  
   SET RDEVICE command [1627](#)  
   support for advanced function printers [1629](#)  
   turning off hot I/O rate [1517](#)  
   UNLOCK command [2044](#)  
   using FOLD option of START UR command [1871](#)  
   using specific volume identifiers for CP-owned volumes [774](#)  
 WARNING command [2097](#)  
 WCGM2 operand  
   SET RDEVICE command [1669](#)  
 WCGM4 operand  
   SET RDEVICE command [1669](#)  
 WEST operand  
   DEFINE TIMEZONE command [225](#)  
 WITH operand  
   ATTACH command [35](#)  
 working allegiance  
   displaying the status of [1372](#)  
   turning simulation on or off [1785](#)

## X

XAEREPSD RECORD file [2220](#)  
 XAUTOLOG command [2100](#)  
 XLINK CHECK command [2110](#)  
 XLINK command [2108](#)  
 XLINK DISPLAY utility [2257](#)  
 XLINK FINDTAB command [2111](#)  
 XLINK FORMAT utility [2260](#)  
 XLINK RESET command [2112](#)

## Z

zone, time  
   changing [225](#), [1738](#)  
   defining [225](#)  
   displaying [1191](#), [1222](#)  
   redefining [225](#)  
   setting [1738](#)







Product Number: 5741-A09

Printed in USA

SC24-6268-73

