

z/VM  
7.3

*Directory Maintenance Facility  
Commands Reference*



**Note:**

Before you use this information and the product it supports, read the information in [“Notices” on page 315.](#)

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: 2022-09-07

© **Copyright International Business Machines Corporation 1979, 2022.**

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

---

# Contents

- Tables..... vii**
  
- About This Document.....ix**
  - Intended Audience..... ix
  - Where To Find More Information..... ix
  - Links to Other Documents and Websites..... ix
  
- How to Send Your Comments to IBM.....xi**
  
- Summary of Changes for z/VM: Directory Maintenance Facility Commands**
  - Reference..... xiii**
    - GC24-6281-73, z/VM 7.3 (September 2022)..... xiii
    - GC24-6281-04, z/VM 7.2 (December 2020).....xiii
    - GC24-6281-03, z/VM 7.2 (September 2020)..... xiii
    - SC24-6281-02, z/VM 7.1 (January 2020).....xiv
    - SC24-6281-01, z/VM 7.1 (April 2019).....xiv
    - [VM66223] Virtual Switch Priority Queuing Enhancement.....xiv
    - SC24-6281-00, z/VM 7.1 (September 2018).....xiv
  
- Chapter 1. Introduction..... 1**
  - Compatibility with Prior DirMaint Support..... 1
  - Using the DirMaint HELP Facility..... 1
  - A Few Words About Defaults..... 2
  - Syntax, Message, and Response Conventions..... 2
  
- Chapter 2. Issuing Commands to DirMaint.....5**
  - Using the Prefix Keywords..... 5
  - DIRMAINT EXEC..... 6
  
- Chapter 3. The DIRMAINT Operands..... 13**
  - ACCOUNT..... 14
  - ACIGROUP.....16
  - ACNTADD..... 17
  - ACNTDEL..... 19
  - ADD.....20
  - AMDISK..... 25
  - APPCPASS..... 31
  - AUTHBY..... 33
  - AUTHFOR..... 34
  - AUTHLINK..... 35
  - AUTHSCIF..... 37
  - AUTOLOG..... 38
  - BACKUP..... 40
  - BATCH..... 42
  - CHECK..... 44
  - CHKSUM..... 46
  - CHNGID..... 47
  - CHVADDR..... 49
  - CLASS..... 50

CLEANUP.....	51
CLONEDISK.....	54
CMDISK.....	59
CMS.....	65
COMMAND.....	66
CONSOLE.....	67
CP.....	69
CPU.....	70
CRYPTO.....	72
DASD.....	74
DASDOPT.....	79
DATAMOVE.....	81
DATEFORM.....	83
DEDICATE.....	84
DEFAULTS, GLOBALV.....	86
DEFINESTAG.....	90
DIRECT.....	92
DIRECTORY.....	93
DIREDIT.....	95
DIRMAP.....	96
DISABLE.....	98
DISTRIB.....	99
DLINK.....	100
DMDISK.....	101
DROPBY.....	103
DROPFOR.....	104
DROPSCIF.....	105
DSECUSER.....	106
DUMP.....	107
D8ONECMD.....	108
ELINK.....	109
ENABLE.....	111
EXECDROP.....	112
EXECLOAD.....	113
EXTNCHK.....	114
FILE.....	115
FREEXT.....	117
GET.....	120
GETCONSOLE.....	121
GLOBALOPTS.....	122
GLOBALV, DEFAULTS.....	124
HELP.....	128
HISTORY.....	130
INCLUDE.....	131
INVEN.....	132
IOPRIORITY.....	133
IPL.....	134
IUCV.....	136
IVP.....	139
LINK.....	144
LOADDEV.....	147
LOCK.....	149
LOGMSG.....	150
LOGONBY.....	151
MACHINE.....	152
MAIL.....	154
MAXSPOOL.....	155
MAXSTORAGE, MAXSTORE.....	156

MDAUDIT.....	158
MDISK.....	162
MDPW.....	165
MINIOPT.....	166
MMDISK.....	168
NAMESAVE.....	170
NEEDPASS.....	171
NEWS.....	172
NICDEF.....	173
NOPDATA.....	177
NOTAPE.....	178
OFFLINE.....	179
ONLINE.....	180
OPTION.....	181
PAVALIAS.....	186
POOL.....	188
POSIXFSROOT.....	189
POSIXGLIST.....	190
POSIXGROUP.....	192
POSIXINFO.....	193
POSIXIUPGM.....	195
POSIXIWDIR.....	196
POSIXOPT.....	197
PRIORITY.....	199
PRIOSET.....	200
PRIVCLASS.....	201
PURGE.....	202
PW.....	204
PW?.....	206
PWGEN.....	207
PWMON.....	209
QLOG.....	211
QUERY, QRY.....	212
REPLACE.....	213
REVIEW.....	215
RLDCODE.....	216
RLDDATA.....	217
RLDXTN.....	218
RMDISK.....	220
SATELLITE.....	224
SCAN.....	226
SCREEN.....	230
SECUSER.....	232
SEND.....	234
SETACNT.....	235
SETCLASS.....	237
SETCPU.....	238
SETMACH.....	240
SETOPTN.....	242
SETPRIORITY.....	247
SETPW.....	248
SETSTAG.....	250
SHARE.....	252
SHUTDOWN.....	254
SPECIAL.....	255
SPOOL.....	258
SSI.....	260
STAG.....	261

STATUS.....	263
STDEVOPT.....	265
STORAGE.....	266
SUBSCRIBE.....	268
SUPGLIST.....	270
SYSAFFIN.....	272
TERM.....	273
TESTPW.....	275
TMDISK.....	277
UNDOSSI.....	279
UNLOCK.....	280
USEDEXT.....	281
USER.....	284
USERMAP.....	286
USEROPTN.....	287
VMRELOCATE.....	291
WORKUNIT.....	293
XAUTOLOG.....	295
XCONFIG.....	297
<b>Appendix A. DirMaint Command Sets.....</b>	<b>299</b>
<b>Appendix B. Device Characteristics.....</b>	<b>307</b>
<b>Appendix C. The Synchronous Application Programming Interface (SAPI).....</b>	<b>311</b>
Sample Programs.....	311
Applied SAPI Coding Rules.....	311
<b>Notices.....</b>	<b>315</b>
Programming Interface Information.....	316
Trademarks.....	316
Terms and Conditions for Product Documentation.....	316
IBM Online Privacy Statement.....	317
<b>Bibliography.....</b>	<b>319</b>
Where to Get z/VM Information.....	319
z/VM Base Library.....	319
z/VM Facilities and Features.....	320
Prerequisite Products.....	322
Related Products.....	322
<b>Index.....</b>	<b>325</b>

---

# Tables

1. Examples of Syntax Diagram Conventions.....	2
2. DirMaint default command sets.....	299
3. Maximum Sizes and Blocks Per Cylinder for Supported CKD Devices.....	307
4. Specifications for a 3390 used in 3380 compatibility mode.....	307
5. Specifications for a Multiprise 2000 internal disk (or any other DASD) emulating a 3380.....	308
6. Specifications for a Multiprise 2000 internal disk (or any other DASD) emulating a 3390.....	308
7. Maximum Sizes and FB-512 Blocks Per CMS Block for Supported FBA Devices.....	308
8. Maximum Sizes and Blocks Per Cylinder for Obsolete CKD Devices.....	309
9. Maximum Sizes and FB-512 Blocks Per CMS Block for Obsolete FBA Devices.....	309



## About This Document

---

This is a reference document for the z/VM® Directory Maintenance Facility (DirMaint™) function level 730, for use on IBM® z/VM 7.3.

The DIRMAINT command operands are listed alphabetically so you can look up their syntax, descriptions, options, and usage notes. The DIRMAINT command and the prefix command descriptions appear before the DIRMAINT command operands.

This document should be used in conjunction with *z/VM: CP Planning and Administration*.

## Intended Audience

---

This document is meant for those persons responsible for creating and maintaining the VM directory. Some functions in this document are also available to general users, allowing them to implement limited changes to their own directory entries.

## Where To Find More Information

---

For a list of the document that can provide you with additional information on DirMaint and z/VM, see “Bibliography” on page 319.

For further up-to-date information specific to DirMaint, see the official DirMaint website at [IBM Directory Maintenance for z/VM \(DirMaint\)](https://www.ibm.com/vm/related/dirmaint) (<https://www.ibm.com/vm/related/dirmaint>)

## 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 Send Your Comments to IBM

---

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

To send us your comments, go to [z/VM Reader's Comment Form \(https://www.ibm.com/systems/campaignmail/z/zvm/zvm-comments\)](https://www.ibm.com/systems/campaignmail/z/zvm/zvm-comments) and complete the form.

## **If You Have a Technical Problem**

Do not use the feedback method. Instead, do one of the following:

- Contact your IBM service representative.
- Contact IBM technical support.
- See [IBM: z/VM Support Resources \(https://www.ibm.com/vm/service\)](https://www.ibm.com/vm/service).
- Go to [IBM Support Portal \(https://www.ibm.com/support/entry/portal/Overview\)](https://www.ibm.com/support/entry/portal/Overview).



# Summary of Changes for z/VM: Directory Maintenance Facility Commands 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.

## GC24-6281-73, 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 command is updated:

- [“DIRECTORY” on page 93](#)

## GC24-6281-04, z/VM 7.2 (December 2020)

---

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

### DirMaint Health Checker

With the PTF for APAR VM66175, the Directory Maintenance Facility (DirMaint), function level 720, includes enhancements that reduce the service cost of the product to the customer. These enhancements can prevent future problems, detect current problems, and reduce the complexity of interacting with DirMaint to perform debugging operations. These enhancements include the following:

- The automated install verification procedure (IVP) runs tests to validate the DirMaint servers and their interaction with each other and the rest of z/VM.
- Improvements to the DIRM FILE command provide additional file protection so that well known files are not accidentally placed on an incorrect disk.
- Changes to the DIRM BACKUP function allow you to backup files that are not on the primary directory files disk, such as CONFIG\* DATADVH files. Some files are allowed on other disks but are not backed up.

The following command is new: [“IVP” on page 139](#).

The following commands are updated:

- [“DATAMOVE” on page 81](#)
- [“DASD” on page 74](#)
- [“FREEEXT” on page 117](#)
- [“SATELLITE ” on page 224](#)
- [“USEDEXT” on page 281](#)

## GC24-6281-03, z/VM 7.2 (September 2020)

---

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

## NEEDPASS\_RESPECT Configuration Option

The Directory Maintenance Facility (DirMaint), function level 720, includes a new configuration option, NEEDPASS\_RESPECT= NO|YES. When the value is NO, DirMaint will no longer prompt for passwords. This is the default value. For compatibility, the NEEDPASS\_RESPECT configuration option can be set to YES. Specifying YES indicates to DirMaint that password prompting should occur based on the user's DIRM NEEDPASS command setting.

## SC24-6281-02, z/VM 7.1 (January 2020)

---

This edition includes changes to support product changes provided or announced after the general availability of z/VM 7.1.

### Improved information about defining crypto resources to z/VM guests

A new chapter has been added to *z/VM: CP Planning and Administration* to provide improved information about how to define crypto resources to z/VM guests. See *Crypto Planning and Management* for more information.

Crypto terminology in z/VM has changed to eliminate confusion.

## SC24-6281-01, z/VM 7.1 (April 2019)

---

This edition includes changes to support product changes provided or announced after the general availability of z/VM 7.1.

### [VM66223] Virtual Switch Priority Queuing Enhancement

With the PTF for APAR VM66219, z/VM support is enhanced to improve data transmission from a virtual NIC to a physical network via the virtual switch uplink port. Internal virtual switch data transmission is changed from a first-in, first-out (FIFO) order to a four-level priority queuing scheme. By default, the virtual switch distinguishes between virtual switch and virtual NIC transmissions to the physical network. Virtual switch transmissions required to manage an uplink port or virtual NIC are sent to the physical network ahead of any virtual NIC transmissions. This insures that virtual switch management is always maintained during high bandwidth uplink port usage.

Virtual switch support is added to exploit the OSA-Express feature's priority queuing capabilities. By turning on virtual switch priority queuing, instead of activating a single QDIO output queue on the uplink port, the z/VM virtual switch will establish four QDIO output queues for outbound data transmissions. The four QDIO output queues provide four independent interfaces to the OSA-Express feature, one for each priority. The highest priority queue (Queue 0) is used by the z/VM virtual switch to provide switch management communications and the other three priorities (Queues 1-3) can be configured by a system administrator to prioritize outbound transmissions to the physical network for virtual NICs connected to the virtual switch.

The NICDEF operand of the DIRMAINT command has been updated.

## SC24-6281-00, z/VM 7.1 (September 2018)

---

This edition includes changes to support the general availability of z/VM 7.1.

### Foundational Support for More Than 64 Logical Processors

Foundational support is provided for a follow-on 7.1 deliverable that will increase the number of supported logical processors beyond the current limit of 64. This will allow clients to run a z/VM LPAR with more than 64 cores or threads to accommodate workload growth demands.

Support for dedicating processors to guests is disabled.

The following sections are updated:

- [“CPU” on page 70](#)
- [“SETCPU” on page 238](#)



---

# Chapter 1. Introduction

DirMaint provides support for all the z/VM directory statements. Most of the DirMaint directory commands have the same names and format as the VM directory statements they support. DirMaint also provides additional utilities to help manage minidisk assignments and allocations, and provide a level of security regarding command authorizations and password monitoring.

When DirMaint is first installed, certain default command authorizations exist. The initial command authorizations are shown in [Appendix A, “DirMaint Command Sets,”](#) on page 299. However, these authorizations are completely tailorable by the installation and therefore are not necessarily the authorizations that currently exist. To find what your current DirMaint authorizations are, consult with your DirMaint administrator.

---

## Compatibility with Prior DirMaint Support

As a general rule, earlier DirMaint function levels will be almost fully command compatible with later DirMaint function levels. There may be isolated changes to the command sets – for instance, when the SPECIAL command was moved from command set A (administrator) to command set G (general user) between releases 4.0 and 5.0.

The earlier function level may still support the earlier command set if that command level is provided by the current release for compatibility with programs designed to work with the earlier DirMaint release. This support is not intended for use by interactive users. Except for any information in Appendix B showing the earlier command sets associated with each command, no documentation of the earlier command level syntax or function will be provided. If your installation had the earlier release installed, you can try issuing `HELP DIRM command_name`, the results of which may still show the earlier release documentation – unless your system programmer has specifically replaced them with the newer HELP files.

Regardless of command level, messages and return codes from earlier function levels will remain compatible with the current release of DirMaint.

---

## Using the DirMaint HELP Facility

You can receive online information about the DirMaint operands described in this book using the DirMaint HELP Facility. For example, to display a menu of DirMaint HELP information, type:

```
DIRM HELP
```

Place the cursor under a command or topic you want information about and press enter.

To display information about a specific DirMaint operand (ADD in this example), type:

```
DIRM HELP ADD
```

or

```
DIRM HELP AD
```

or

```
DIRM HELP A
```

The DirMaint HELP facility recognizes the minimum abbreviation for a DIRMAINT operand.

To display information about the DIRMAINT command, type:

```
DIRM HELP DIRM
```

You can also display information about a message (DVH1093 in this example), by typing:

```
DIRM HELP DVH1093
```

The DIRMAINT HELP command and its operands are described in [“HELP” on page 128](#).

## A Few Words About Defaults

For DirMaint directory commands, any defaults that you assume when issuing the command will be the same as the defaults in place for the release of VM installed on your system. Because of this, the VM defaults are not shown in the syntax diagrams. However, any defaults set by DirMaint are shown.

Ensure that you have a copy of *z/VM: CP Planning and Administration* for the appropriate VM version and release installed on your system to reference the directory statement default settings.

## 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  $\blacktriangleright\blacktriangleright$ — 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 — $\blacktriangleright\blacktriangleleft$  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 1 on page 2](#).

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 1. Examples of Syntax Diagram Conventions (continued)

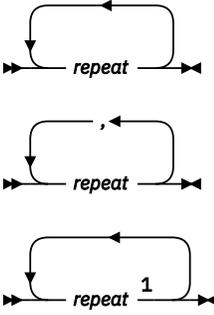
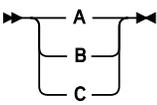
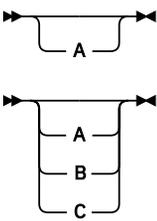
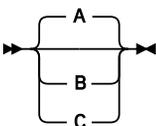
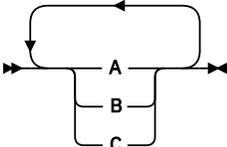
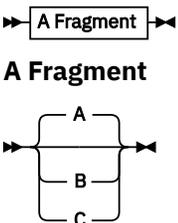
Syntax Diagram Convention	Example
<p><b>Symbols</b></p> <p>You must specify these symbols exactly as they appear in the syntax diagram.</p>	<p>* Asterisk</p> <p>:</p> <p>Colon</p> <p>,</p> <p>Comma</p> <p>=</p> <p>Equal Sign</p> <p>-</p> <p>Hyphen</p> <p>()</p> <p>Parentheses</p> <p>.</p> <p>Period</p>
<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 1. 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.

---

## Chapter 2. Issuing Commands to DirMaint

The DIRMAINT command provides the interface between the command issuer and the DIRMAINT service machine. The issuer types the "DIRMAINT" command followed by a command operand. The DIRMAINT command routes the command to the DIRMAINT service machine where the service machine does validation checking and either processes the request or rejects it with an appropriate message. The request sent to the service machine may include prefix keywords. For example:

```
EXEC DIRMAINT REVIEW
```

or

```
DIRM TONODE CHQ1 AUTHBY ADMIN1
```

---

### Using the Prefix Keywords

Additional instructions can be given to the DIRMAINT service machine regarding the processing of a command operand by using a prefix keyword. The prefix keyword (and any operands required for the keyword) is placed before the operand whose processing it is to affect. For example:

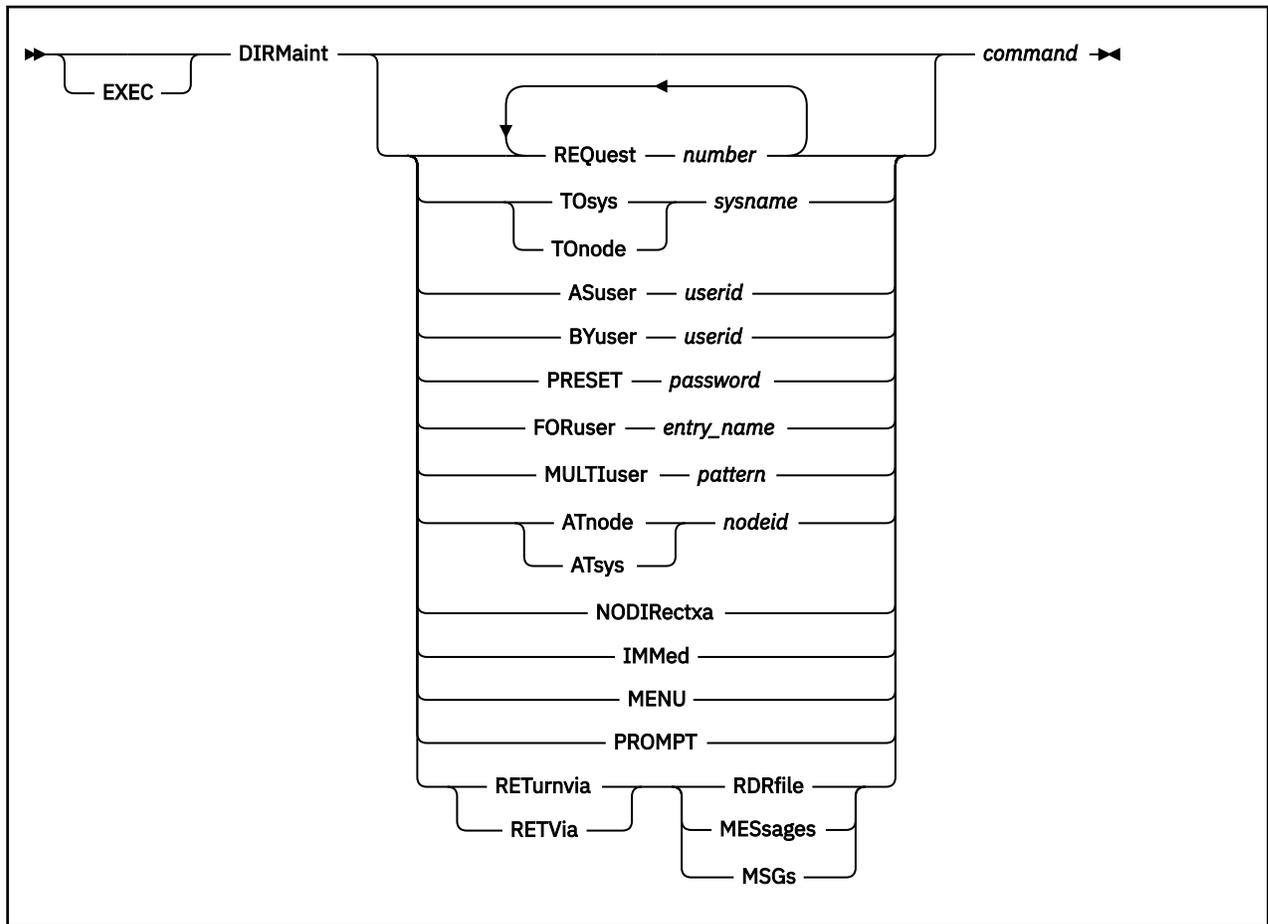
```
DIRM AT SYSTEMA MACHINE ESA
```

Multiple prefix keywords can be used in a single command string, but each prefix keyword can only be specified once. For example:

```
DIRM TO CHQ1 AS MAINT BY ADMIN1 FOR CEO AT CHQ2 IMMED IPL CMS2
```

Because some prefix keywords allow DirMaint commands to be issued on behalf of another user ID's authority, users must be authorized to use these prefix keywords. Not all DIRMAINT commands accept prefix keywords. Those that do or do not are noted in the "Usage Notes" section of the command description. Prefix keywords and their usage are described under the description of the DIRMAINT command.

## DIRMAINT EXEC

**Purpose**

Use the DIRMAINT EXEC to provide the command interface between the user and the DIRMAINT virtual machine. The DIRMAINT EXEC sends DirMaint commands to the DIRMAINT service virtual machine for execution.

**Operands****EXEC**

is optional when the DIRMAINT EXEC is invoked from the CMS command line at the *Ready;* prompt, or when issued within a REXX exec with ADDRESS CMS specified. It is required when the DIRMAINT EXEC is invoked from certain subsystems, or when issued within a non-REXX exec, or within a REXX exec with ADDRESS COMMAND specified.

**DIRMaint**

is the name of the exec. It may be abbreviated to DIRM unless it is prefaced with the EXEC keyword, or the issuing user is running with SET ABBREV OFF or SYNONYM (NOSTD).

**REQuest number**

associates a numeric value in the range of 1 to 999999 with a specific DirMaint transaction.

This prefix operand is intended to be used primarily with the SAPI interface.

**TOsys sysname****TOnode sysname**

specifies the network node name of the remote system to which the command will be sent for execution.

If the TO parameter is not specified, the default may have been previously set using the DIRMAINT DEFAULTS operand. If the value of the TO parameter is an asterisk (\*), either by default or explicit request, the command will be processed within the local system cluster.

The following entries must be defined in the CONFIG\* DATADVH file to identify the remote system that TOsys (or TOnode) is to send the DIRMAINT request, unless the IDENTIFY command correctly identifies the network traffic machine, and the user ID of the DIRMAINT service machine on the remote system is DIRMAINT:

```
FROM= fromnode DEST= tonick S= spoolid T= destnode2 U= serverid
```

Where:

***fromnode***

identifies the network node ID where the transaction originates.

***tonick***

identifies the nickname for the remote system where the transaction is being sent.

***spoolid***

identifies the user ID of the machine where punch output should be sent to reach the specified destination. For use within the local system cluster with cross system spooling enabled, this is the user ID of the DIRMAINT service machine; otherwise it is the user ID of an RSCS network service machine.

***destnode***

identifies the remote system where the transaction is to be processed, as that system is known to the network.

***serverid***

identifies the user ID of the DIRMAINT service machine on the remote system.

If no entry is found for the specified *sysname* as either a DEST= *nickname* or as a T= *destnode*, then a second lookup is attempted using a node ID of asterisk (\*). If the *spoolid* is an \*, then the network service machine's user ID is obtained using the CMS IDENTIFY command. If the *destnode* tag field is an \*, then the specified *sysname* is used. If the *serverid* is an \*, then the name DIRMAINT is used.

**Note:** TOSYS and TONODE are synonymous.

**ASuser *userid***

specifies the user ID under whose privileges a command will be performed. This keyword is primarily intended for use with the TOsys keyword. When the NEEDPASS\_RESPECT configuration option is either allowed to default to NO or is specifically set to NO, passwords are not necessary when executing DirMaint commands. However, when the NEEDPASS\_RESPECT configuration option is set to YES and the user's NEEDPASS setting is YES, unless the BY keyword is also specified the password is verified for the user ID issuing the command. Also, use of the AS parameter causes a password to be required for authentication, even if you have NEEDPASS set to NO and even if the ASuser user ID has specified NEEDPASS NO on the target system.

If the AS parameter is not specified, the default may have been previously set using the DIRMAINT DEFAULTS operand. If the value of the AS and BY parameters are both an asterisk (\*), either by default or explicit request, any password verification will be done with the user ID from which the command was issued.

The user ID specified with the AS keyword must exist on the target system where the command is to be executed.

If the value of the FOR keyword is an asterisk (\*) after default resolution, and the value of the AS keyword is not an \*, then the value of the AS parameter will also be used as the value of the FOR keyword.

**BYuser *userid***

specifies the user ID that is issuing the command. This keyword is not required when the NEEDPASS\_RESPECT configuration option is set to NO (the default) and the user ID issuing the command is a shared virtual machine that has been logged onto by one of the sharing users using

their personal password. When the NEEDPASS\_REPSECT configuration option is set to YES this keyword is required when the user ID issuing the command is a shared virtual machine that has been logged onto by one of the sharing users using their personal password, and that user does not know the current logon password for the shared virtual machine. Unless the AS keyword is specified, the command is executed with the privileges of the command issuer. The BYUSER keyword is unnecessary if the user issuing the command has set NEEDPASS NO, or if the user issuing the command knows the logon password of the user ID where the command is issued. Use of the BY parameter causes a password to be required for authentication, even if you have NEEDPASS NO and even if the BYuser user ID has specified NEEDPASS NO on the target system.

If the BY parameter is not specified, the default may have been previously set using the DIRMAINT GLOBALV operand. If the value of the AS and BY parameters are both an asterisk (\*), either by default or explicit request, and the NEEDPASS\_RESPECT configuration option is set to YES, password verification will be done for the user ID from which the command was issued.

The issuing user ID (or the AS user ID) and the user ID specified with the BY keyword must both exist on the target system where the command is to be executed. The issuing user ID (or the AS user ID) must have previously issued an AUTHBY command for the BY user ID. When the NEEDPASS\_RESPECT configuration option is set to YES, the password must be the correct logon password for the BY user ID on the target system.

### **PRESET password**

specifies the password or passphrase you would enter in response to message DVHXMT1181R. This avoids the prompt. When entering the PRESET password or passphrase on the VM command line, note that a passphrase containing embedded blanks must be surrounded by single quotes (with single quote characters within the passphrase itself doubled). When entering the PRESET password or passphrase on the prefix menu panel, up to 40 characters may be used. Longer passphrases must be specified by entering the DIRMaint command on the VM command line or by issuing the DirMaint command in a z/VM Rexx EXEC. The password or passphrase will not appear in DirMaint maintained log files.



**Attention:** Use of this parameter may show up in console trace logs if tracing is turned on.

### **FORuser entry\_name**

specifies the name of the directory entry to be changed. This keyword is optional for all of the general user commands, but is required for most of the privileged user commands.

If the FOR parameter is not specified, the default may have been previously set using the DIRMAINT GLOBALV operand. If the value of the FOR parameter is still an asterisk (\*) after default resolution, and the value of the AS keyword is not an \*, then the value of the AS parameter is also used as the value of the FOR keyword. If the value of the FOR keyword is still an \*, either by default or explicit request, any directory change affects the user ID that issued the command.

When issued by a general user, the user ID associated with the entry name specified with the FOR keyword must have previously issued an AUTHFOR command to authorize the issuer (or the AS user ID).

### **MULTIuser**

identifies one or more entries (through the use of a scan pattern) to be acted upon by the specified command. The pattern supplied with the MULTIUSER prefix operand is used to build a list of entries. Once the list has been built, a batch file is generated to do the actual commands. Each batch command is issued using the invoker's authority. The pattern uses a *wild card* character in much the same way that the CMS LISTFILE command uses it. The following rules apply:

- A single asterisk (\*) matches all entries on the system.
- Multiple, adjacent asterisks are considered to be a single \*, (for instance, ABC\*\*\*123 is equal to ABC\*123).
- Entries \$DIRCTL\$, \$DIRGRP\$, ALL, and SYSTEM are always excluded from matches and are never the target of a MULTIUSER command.
- Leading, trailing, and imbedded asterisks may be used in a pattern.

- Multiple asterisks may be used within a single pattern (for instance, H\*WL\*ND\* or \*O\*A\*).

Users of this prefix operand must be authorized by the MULTIUSER\_VERIFICATION\_EXIT. This exit controls what users are authorized to use the MULTIUSER prefix operand. In addition, this prefix operand can not be used with some DIRMAINT commands (the specific commands are listed in the sample exit routine). This exit routine must be altered by the local installation to enable this support. As shipped, the exit routine rejects all transactions.

The MULTIUSER\_VERIFICATION\_EXIT also has the ability to provide a list of entries (overriding the pattern matching) for use with the provided command. This allows the local site to use whatever method is appropriate to subset the entries contained in the USER DIRECT file. Details on the steps required to utilize this capability are documented in the sample exit routine.

If this prefix is used with the FORUSER prefix operand, the FORUSER operand is ignored.

### **ATnode *nodeid***

#### **ATsys *nodeid***

specifies the system node name within a multiple system cluster that the directory change is to affect, without affecting other nodes in the cluster. If the directory statement to be affected already exists in the directory with internal SYSAFFIN statements that conflict with this request, the command will be rejected by the DIRMAINT service machine. If the AT keyword is not specified, the default may have been previously set using the DIRMAINT GLOBALV operand.

In a CSE cluster environment, if the value of the AT keyword is an asterisk (\*), either by default or explicit request, then any directory change will affect all nodes within the cluster on which the command is processed, unless the directory statement associated with the change is overridden for a specific system affinity. In order to change a directory statement that applies to a certain system affinity, you must supply the specific system affinity using the AT prefix.

In a single system image (SSI) environment, the AT prefix keyword will be ignored if it is specified with a DirMaint command for a user or profile entry. If the AT prefix is an asterisk (\*), either by default or explicit request, then the directory change will affect the members to which the related directory entry applies. If the change is for an identity entry, the change will affect all members in the cluster. If the change is for a subconfiguration entry, the change will affect only the member associated with the subconfiguration entry. If a specific member is supplied in the AT prefix for an identity entry, its value will specify the system for which the identity's subconfiguration entry will be modified. If AT is specified for an identity or subconfiguration entry when a DASD management command has been issued, the AT keyword value will be used to select a DATAMOVE machine configured for the specified member. In this case, the FORuser prefix keyword must specify the subconfiguration entry name to modify a subconfiguration entry with the following DASD management commands: AMDISK, CLONEDISK, CMDISK, DMDISK, MMDISK, RMDISK, TMDISK and PURGE. Otherwise, the AT prefix is ignored.

**Note:** ATNODE and ATSYS are synonymous.

### **NODIRECTXA**

The NODIRECTXA prefix specifies that DIRECTXA will not be called to put the object directory online, even if ONLINE=IMMEDIATE has been specified. DirMaint will behave as if ONLINE=SCHED has been specified for this one request. (If the change can be put online using Diagnose X'84', it will be.)

### **IMMed**

The IMMED prefix specifies that the DirMaint request will be sent as a RDR file, rather than as an SMSG. This is useful:

- When the DIRMAINT server needs to communicate with other non-DirMaint servers, such as RACF or other programs that pre-empt SMSG communications
- To ensure that multiple requests sent to the DIRMAINT server are processed in the order sent when those that use a RDR file out of necessity (such as ADD without LIKE, BATCH, FILE, or PWMON) are mixed with other requests.

### **MENU**

specifies that a menu be displayed (if available) for the specified command keyword prior to its execution.

Any parameters provided beyond the command keyword will be ignored by the menu processor. The command string will be built by the menu processor and passed to DirMaint for evaluation and execution.

The MENU prefix operand is implied if the command string given to DirMaint requires parameters beyond the command key word but they are not provided.

Other prefix operands can be specified in conjunction with the MENU prefix operand and are kept/held/used by the command built by the menu processor. For example:

```
DIRM FOR USERAA MENU PW
```

If a menu is not available for the command keyword, the command string will be passed to DirMaint for evaluation and execution as entered. Menus are not provided for the following DirMaint commands:

- ?
- CLEANED
- COPIED
- VERIFIED

**PROMPT**

requests that password prompting be done if passwords are not provided on the command line for specific operands of the DIRMAINT command.

This prefix operand is accepted on all commands but is primarily intended for use with the APPCPASS, LINK, and MDISK operands of the DIRMAINT command.

**RETurnvia RDRfile|MESsages|MSGs****RETVia RDRfile|MESsages|MSGs**

If specified with RDRfile, causes DVHSEND to return the data as a spool file. If specified with either MESsages or MSGs, causes DVHSEND to return the data as DVHSND3792I messages. If not specified, the default is RETURNVIA RDRFILE.

**Note:** RETurnvia and RETVia are synonymous. MESsages and MSGs are also synonymous.

***command***

any DirMaint command.

**Usage Notes**

1. One or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa, MENU, PROMPT, RETurnvia/RETVia) may be specified in any order, but each keyword may only be specified once.
2. Most DirMaint commands are sent to the DIRMAINT virtual machine for processing. Passwords are not necessary when executing DirMaint commands when the NEEDPASS\_RESPECT configuration option is either allowed to default to NO or is specifically set to NO. When the NEEDPASS\_RESPECT configuration is set to YES, you will be prompted to enter a logon password for authentication unless you have issued a DIRM NEEDPASS NO command. If the AS or BY keyword parameters are specified, you will be prompted to enter a logon password for authentication regardless of the value of the NEEDPASS option. If you are running as a disconnected service machine, you may stack the password before invoking the DIRMAINT EXEC, or may set the password in the PRESET global variable on the DVH15 variable pool.
3. If the PRESET global variable, in the DVH15 pool, has a value other than blanks or NOLOG, DirMaint will send that value to the DIRMAINT server without prompting for a password. To prevent disclosure of that password, the variable will be reset to a value of NOLOG upon exit, thus the value must be set prior to each call to the DIRMAINT EXEC.
4. There are a few DirMaint commands that are completely processed within the virtual machine of the command issuer. No transaction is sent to the DIRMAINT service machine. Therefore, no password is required or requested for authentication, even if you have set NEEDPASS to YES. If you are running as

a disconnected service machine, nothing will be pulled from the stack. The following commands are completely processed within the issuer's virtual machine:

- CHECK
  - DEFAULTS
  - EXECDROP
  - EXECLOAD
  - GLOBALV
  - HELP
  - any command with an invalid syntax
5. The NEEDPASS command is an exception to the PREFIX KEYWORDS, ignoring any PREFIX keyword that may have been supplied on the command.
  6. The DIRMAINT command itself is not valid within a BATCH command file. Some of the prefix operands may or may not be used within a BATCH command file, or as prefixes to a BATCH command, as follows:
    - The ASuser, and BYuser prefix operands are not valid within a BATCH command file. If used, the request will receive error 2218E. Either operand may be used, however, as a prefix to a BATCH command.
    - The TOsys prefix may be used on a request within a BATCH command file, but the specified node must be the local system. Otherwise, the request will receive error 2291E. TOsys may also be used on the BATCH command itself.
    - The FORuser and ATnode prefix operands are valid on requests within a BATCH command file, but they are not valid as a prefix to a BATCH command (in which case they will be ignored).
  7. If you are planning to issue DIRMAINT commands from exec programs and perform actions based upon message responses from the DIRMAINT service machine, see Appendix C, “[The Synchronous Application Programming Interface \(SAPI\)](#),” on [page 311](#) for instructions on how to set up your execs to activate the Synchronous Application Programming Interface (SAPI) and access the response information.

## Examples

### • EXEC

```
EXEC DIRMAINT REVIEW
```

### • REQuest

```
DIRM REQ 6 SECUSER EDWARDSR
```

### • TOsys, TOnode

```
DIRMAINT TO SYSTEM1 ENABLE
DIRM GLOBALV DTO *
DIRM GLOBALV DTOSYS SCHOOL
DIRM TONODE CHQ1 AUTHBY ADMIN1
DIRMAINT TOSYS CHQ1 BY ADMIN1 FOR CEO AT
  CHQ2 IMMED IPL CMS2
  password
DIRM TO SCHOOL FOR * BY * AUTHBY PROF2
```

### • ASuser

```
DIRMAINT ASUSER ALTEREGO ACCOUNT PROJECT1
  password
```

## DIRMAINT

- **BYuser**

```
DIRMAINT AUTHBY OWNER
DIRM BY OWNER ONLINE
password

DIRM TO CHQ1 AUTHBY ADMIN1
DIRM TO CHQ1 BY ADMIN1 FOR CEO AT
CHQ2 IMMED IPL CMS2
password
```

- **FORuser**

```
DIRM AUTHFOR ISSUER
DIRM FOR TARGETID DISTRIB NEWDEPT
```

```
DIRM TO CHQ1 AUTHBY ADMIN1
DIRM TO CHQ1 BY ADMIN1 FOR CEO AT
CHQ2 IMMED IPL CMS2
password
```

```
XEDIT SETPW CLASS101 A
INPUT FOR STUDENT1 SETPW CLASS101
INPUT FOR STUDENT2 SETPW CLASS101
INPUT FOR STUDENT3 SETPW CLASS101
INPUT FOR STUDENT4 SETPW CLASS101
INPUT FOR STUDENT5 SETPW CLASS101
INPUT FOR STUDENT6 SETPW CLASS101
INPUT FOR STUDENT7 SETPW CLASS101
INPUT FOR STUDENT8 SETPW CLASS101
INPUT FOR STUDENT9 SETPW CLASS101
FILE
DIRM BATCH SETPW CLASS101 A
```

- **MULTIuser**

```
DIRM MULTIUSER G32* DIST G32/B90
DIRM MULTIUSER *USER* ACCOUNT NEWACNT
```

- **ATnode ATsys**

```
DIRM AT SYSTEMA MACHINE ESA
```

- **NODIRctxa**

```
DIRM NODIR SPECIAL 100 CTCA
```

- **MENU**

```
DIRMAINT MENU REVIEW
```

- **PROMPT**

```
DIRMAINT PROMPT MDISK 191 MR
```

- **RETurnvia RETVia**

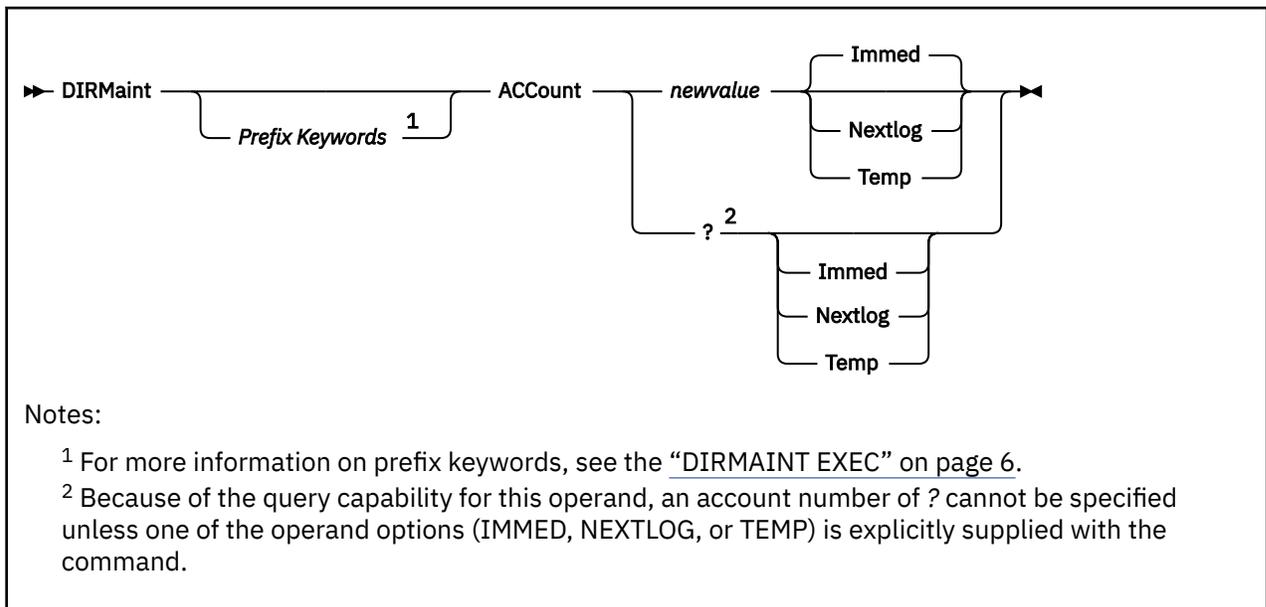
```
DIRM RET MSG SEND EXTENT CONTROL
```

---

## Chapter 3. The DIRMAINT Operands

This section describes the syntax, environment, purpose, operand descriptions, and any additional usage information regarding the DIRMAINT command operands. In some cases, examples of usage are provided.

## ACCOUNT



### Purpose

Use the ACCOUNT operand of the DIRMAINT command to enable you to use an account number other than the one defined as *currently in use* in the ACCOUNT control statement. You can determine the *current* account number by issuing:

```
DIRM ACCOUNT ?
```

If your installation has provided alternate account numbers for you to select from, they must be defined in your directory entry. The method of providing alternate account numbers is a function of the operating system environment and your installation's policy. For more information on defining and preparing alternate account numbers for your installation, see [z/VM: CP Planning and Administration](#).

### Operands

#### **newvalue**

is the new account number to be brought into use for your virtual machine. It must be one of the alternate account numbers defined in your directory entry. To determine the alternate account numbers, issue the DIRM REVIEW command operand to obtain a copy of your directory entry and inspect the appropriate statements. By inspecting the optional \*AC= statement and the secondary account numbers on the ACCOUNT statement you can determine what account numbers are valid for you to use. You may select any one of the 1 to 8 character account numbers defined.

#### **Immed**

specifies that the new account number is to become effective immediately and remain in effect until changed. The source directory is updated to reflect the change. This is the default value.

#### **Nextlog**

specifies that the new account number is to become effective with the next logon and remain in effect until changed. The source directory is updated to reflect the change.

#### **Temp**

specifies that the new account number is to be used only for the current terminal session. The source directory is not changed.

#### **?**

allows you to query your current account number.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For a complete description of the ACCOUNT directory statement, see *z/VM: CP Planning and Administration*.
3. Use of the IMMED and TEMP options require that the CP update-in-place facility has been enabled by the installation. This is done by placing an UPDATE\_IN\_PLACE=YES record in the DIRMAINT configuration file. If this is not the case, the user will be informed through messages. The transaction will not take place.
4. The directory is altered when using the IMMED and NEXTLOG options. The TEMP option updates the account number for the current session only.
5. The \*AC= statement is an optional statement containing one or more valid secondary account numbers. For more information, see “SETACNT” on page 235.
6. When determining the validity of an account number, the directory and its included profile are consulted to build a list of candidate account numbers. The effective ACCOUNT and \*AC= statements are used for this purpose. The effective statement is defined as the first statement found by looking at the directory and then the profile. For example, if the statement occurs in both the user directory and the profile, the user directory statement is used.
7. If a locally installed ACCOUNT\_NUMBER\_VERIFICATION\_EXIT returns control to the ACCOUNT command handler with a return code of zero, no additional checking will be done on the account number, and it will be accepted without consulting the secondary account numbers or the \*AC= statement.
8. When altering an existing account record, the current primary account number is replaced with the new number, provided the new number passed all required validity checks. The current account number is not placed on the secondary list unless the new primary account number was validated by consulting the secondary list. If the current primary account number is not on the \*AC= statement and not on the current list of secondary account numbers, it is destroyed during the replace process and can not be recovered.
9. When altering an account record in a directory where an account record did not exist, the account record in the profile is used (if it exists) as the effective account statement. Any secondary account numbers and the distribution code on the account record from the profile are placed into the user directory entry along with the new primary account number.

For example, if the user directory contains no ACCOUNT statement, but does include the following \*AC= statement:

```
*AC= MARK0001 MIKE0002 CHUCK003
```

And, in addition, the user directory includes a profile that contains the following account statement:

```
ACCOUNT DEPTG32 BOX90C PROJ001 PROJ002 PROJ003
```

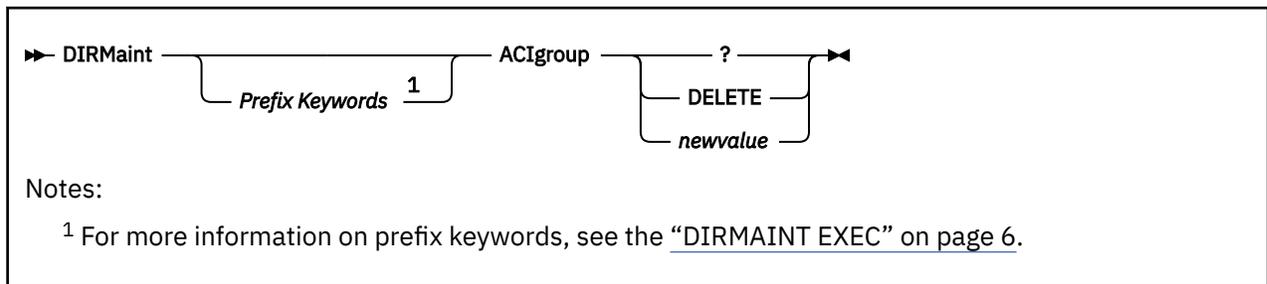
the command:

```
DIRM ACCOUNT MIKE0002
```

will result in the following ACCOUNT statement being placed into the directory entry:

```
ACCOUNT MIKE0002 BOX90C PROJ001 PROJ002 PROJ003
```

## ACIGROUP



### Purpose

Use the ACIGROUP operand of the DIRMAINT command to enable you to specify, change, query or delete the ACIGROUP statement in the current user directory.

The ACIGROUP statement is used to specify the name of the group to which this user is assigned.

### Operands

**?**

allows you to query your ACIGROUP setting.

**DELETE**

allows you to delete your ACIGROUP statement.

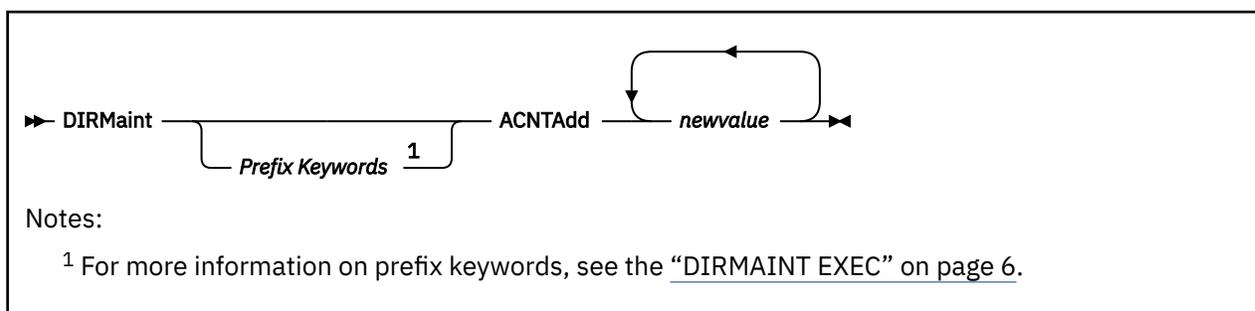
***newvalue***

is the new or changed group name to be associated with this user ID. If specified it will replace any existing value on an existing ACIGROUP statement.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For a complete description of the ACIGROUP directory statement, see [z/VM: CP Planning and Administration](#).
3. Due to the syntax of this command, ? and DELETE are not allowed as ACI group names. If one of these values is required as an ACI group name, use DIRM GET and DIRM REPLACE to set it.

## ACNTADD



### Purpose

Use the ACNTADD operand of the DIRMAINT command to enable you to add an account number to your list of secondary account numbers on your ACCOUNT control statement. You can determine what your current account control record looks like by issuing:

```
DIRM ACCOUNT ?
```

The method of providing alternate account numbers is a function of operating system environment and installation policy. For more information on defining and preparing alternate account numbers for your installation, see *z/VM: CP Planning and Administration*.

### Operands

#### *newvalue*

specifies an account number to be placed on the ACCOUNT control record as a secondary account number. 1 to 7 *newvalue* parameters are allowed. *newvalue* must be one of the alternate account numbers defined in your directory entry, or be specifically authorized by a local exit. To determine the alternate account numbers, issue the DIRM REVIEW command operand to obtain a copy of your directory entry and inspect the appropriate statements. By inspecting the \*AC= statement you can determine what account numbers are valid for you to use. You may select any one of the 1 to 8 character account numbers defined.

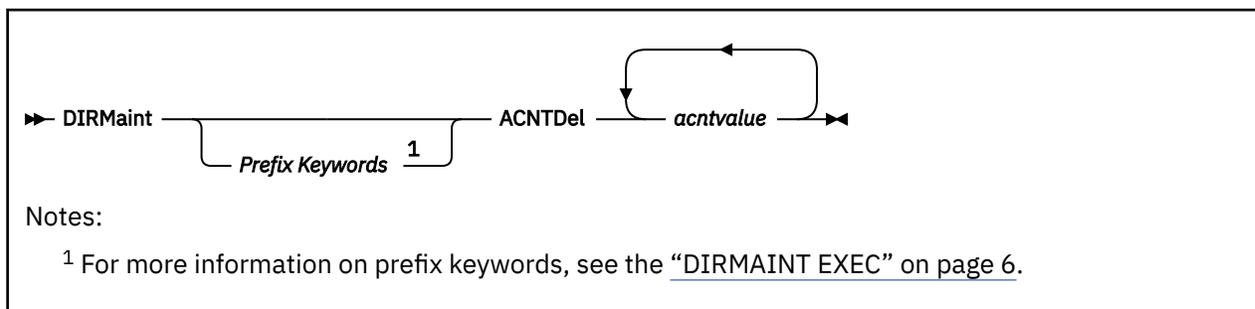
### Usage Notes

1. You may use one or more of the prefix keywords (REQest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. When determining the validity of an account number, the directory and its included profile are consulted to build a list of candidate account numbers. The effective \*AC= statement is used for this purpose. The effective statement is defined as the first statement found by looking at the directory and then the profile. That is, if the statement occurs in the user directory and the profile, the user directory statement is used.
3. If a locally installed ACCOUNT\_NUMBER\_VERIFICATION\_EXIT returns control to the ACNTADD command handler with a return code of 0, no additional checking will be done on the account number and it will be accepted without consulting the \*AC= statement.
4. When adding an account number (or numbers) to the secondary list on the account number, the entire request will be rejected if a number on the add list already exists on the current list of secondary account numbers.
5. If you are using this command to add secondary account numbers to a nonexistent account statement in a user directory, the primary account number and distribution code default to the directory name

after checking the profile for an existing account statement. If the account statement exists but contains only the primary account number, the distribution code defaults to the directory name.

6. If you attempt to place more than seven account numbers on the secondary list, the entire request is rejected. The amount is determined by adding the existing secondary account number and the proposed new account numbers.
7. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message when the source directory has been updated and when the change has been placed online.

## ACNTDEL



### Purpose

Use the ACNTDEL operand of the DIRMAINT command to enable you to remove an account number (or numbers) from your list of secondary account numbers on your ACCOUNT control statement. You can determine what your current account control record looks like by issuing:

```
DIRM ACCOUNT ?
```

### Operands

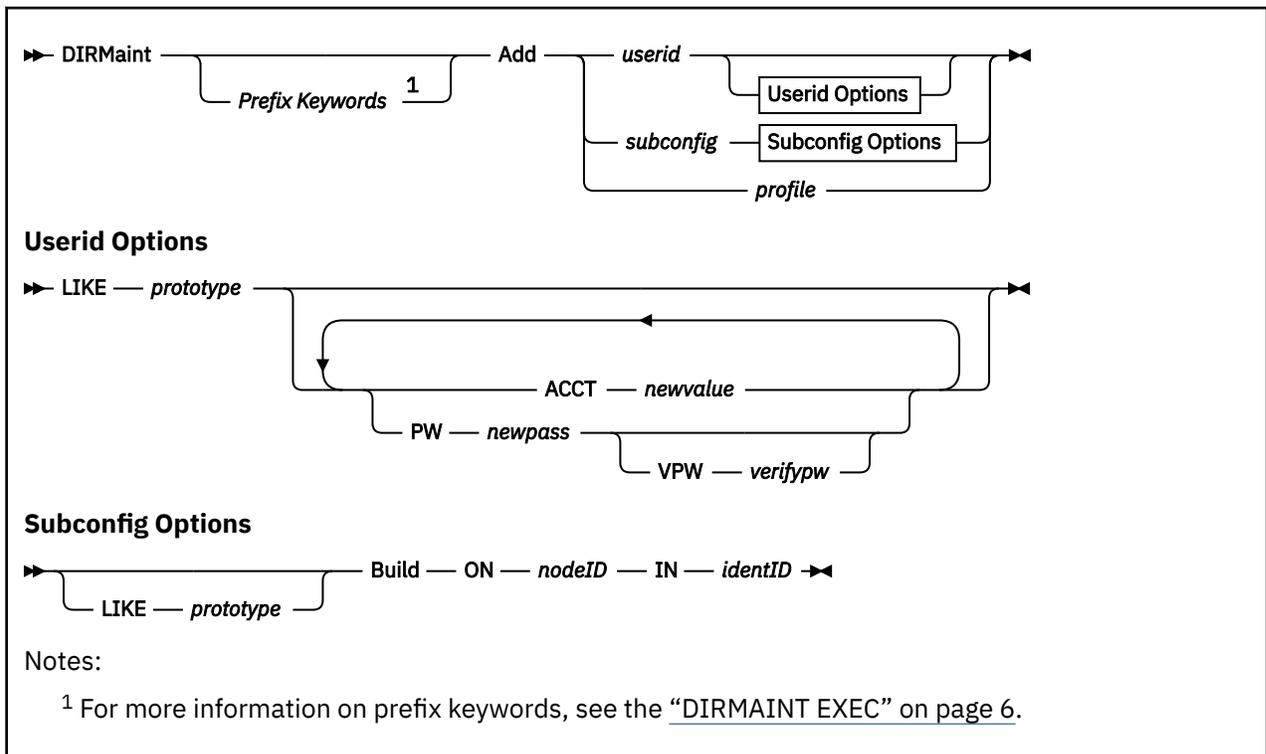
#### *acctvalue*

is a list of 1 to 7 account numbers to be removed from the ACCOUNT control record secondary account number list.

### Usage Notes

1. You may use one or more of the prefix keywords (REQest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. The specified account numbers must exist on the current ACCOUNT control statement. If a specified number does not exist on the control statement the entire request will be rejected.
3. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message when the source directory has been updated and when the change has been placed online.

## ADD



### Purpose

Use the ADD operand of the DIRMAINT command to add an entry to the directory for a new user, identity, subconfiguration, or profile. The configuration of the user, identity, or subconfiguration can be created in the ADD operand invoker's virtual machine and forwarded to DIRMAINT, or the configuration can be taken from a prototype definition stored in the DIRMAINT server. For subconfiguration entries, the configuration can also be taken from an existing subconfiguration entry. Profile entries cannot be added using the LIKE keyword. Use the BUILD option to add a BUILD statement into the specified identity entry for the subconfiguration being added.

### Operands

#### *userid*

identifies the user ID of the new user or identity entry you are creating. The user ID must be the file name of a file with a file type of DIRECT containing the user or identity directory entry being created, unless the LIKE parameter specifies the file name of an existing prototype. In the case of a prototype, the prototype must contain a user entry if you are creating a single-configuration virtual machine or must contain an identity entry if you are creating a multiconfiguration virtual machine.

#### *subconfig*

identifies the ID of the new subconfiguration entry you are creating. The subconfiguration ID must be the file name of a file with a file type of DIRECT containing the subconfiguration directory entry to be built on the SSI member specified in the BUILD option. If the LIKE parameter is specified, the subconfiguration ID specifies either the file name of an existing prototype which contains a subconfiguration entry or specifies the subconfiguration ID of an existing subconfiguration entry in the CP directory that is associated with the identity specified in the BUILD option.

#### LIKE *prototype*

indicates that the new *userid* or *subconfig* is to be added LIKE the specified stored *prototype* file or existing subconfiguration entry.

**PW password**

specifies the logon password to be installed in the directory entry for the new *userid*.

See Usage Note “25” on page 24.

**VPW verifypw**

specifies the same password again for typographical verification.

**profile**

is the name of the new profile entry you are creating.

**ACCT newvalue**

optionally specifies an account number to be installed in the created *userid*.

**Build**

indicates that a BUILD statement should be added in the identity directory entry with the specified *nodeID* for the *subconfig* being created.

**nodeID**

identifies the node ID of the SSI cluster member for the BUILD statement being added to the identity directory entry for the *subconfig* being created.

**identID**

specifies the user ID of the identity directory entry to which a BUILD statement will be added for the *subconfig* you are creating.

**Usage Notes**

1. To delete a directory entry, use the PURGE operand of the DIRMAINT command.
2. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords do not apply to this command. They are accepted without error, but otherwise ignored.
3. Although CP allows a profile entry to be defined in the directory with the same name as a user ID or subconfiguration entry, DirMaint requires that every directory entry (either user ID, subconfiguration, or profile) be unique from all others. Therefore, all directory entries in the CP directory managed by DirMaint must have a unique name/ID in respect to all other CP directory entries.
4. The ADD command is not valid in a BATCH command unless the LIKE keyword is used.
5. The LIKE keyword can be used to create a user, identity, or subconfiguration entry that matches a prototype created by the installation and stored on the DIRMAINT service machine. The specified prototype may be modeled after an existing directory entry (that is, *userid* DIRECT or *subconfig* DIRECT).

When using the LIKE keyword, the DIRMAINT server searches its accessed disks for a prototype definition with a file name of *prototype* and a file type of PROTODIR. This prototype directory entry is then altered with the provided *userid* or *subconfig*, as well as the *newpass* information before being added to the system.

Note that there are IBM-supplied prototypes for typical CMS and LINUX virtual machines. Refer to note “17” on page 23 for installation tips. If a PROTODIR file does not exist and the BUILD option is specified, DirMaint searches the source directory for a subconfiguration entry with the same ID specified in *prototype* and tries to create a new subconfiguration entry based on the existing subconfiguration entry.

6. To create a user ID that is defined by an identity entry:
  - a. Add the identity entry. It should not include any BUILD statements. BUILD statements will be added when the subconfiguration entries are added.
  - b. Add a subconfiguration entry for each member of the cluster on which the user ID will exist. The subconfiguration options that are specified with the ADD operand provide the information needed to create the BUILD statement within the identity entry.

7. You can easily add profiles, user IDs, or subconfigurations by keeping skeletal directory definitions in *profile* DIRECT, *userid* DIRECT, and *subconfig* DIRECT files containing the standard control statements. These files can be copied and the file name changed to the new profile name, user ID, or subconfiguration name you are adding. The new file can then be modified as necessary with the CMS editor. For example, you might keep files like the following two on your own minidisk and use them as models for adding new profiles and single-configuration user IDs. To make a new PROFILE or USER statement, you would insert the new profile name in the PROFILE statement, or the new user ID and CP logon password in the USER statement. Then modify the other fields as appropriate.

```
PROFILE profile
  CONSOLE 009 3215
  SPOOL 00C 2540 R
  SPOOL 00D 2540 P
  SPOOL 00E 1403
  SPECIAL OFF TIMER
  LINK MAINT 190 190 RR
  LINK MAINT 19E 19E RR

USER userid password 4M 16M G 64
  INCLUDE PROFILEX
  ACCOUNT 99999999 distcode
  MDISK 191 XXXX AUTOG 004 GROUPNAM MR
```

When adding a new multiconfiguration user ID, you might have two files like the following (one for the identity entry and one for the subconfiguration entry). To make a new IDENTITY or SUBCONFIG statement, insert the new user ID and CP logon password in the IDENTITY statement, or the new subconfiguration name in the SUBCONFIG statement. Then modify the other fields, as appropriate:

```
IDENTITY userid password 32M 64M G
  INCLUDE PROFILEX
  ACCOUNT 99999999 distcode

SUBCONFIG subconfig
  MDISK 191 XXXX AUTOG 004 GROUPNAM MR
```

8. The AMDISK and MDISK control statements can specify automatic allocation on a specific *valid* by using the AUTOV keyword instead of a starting cylinder or block number as in this example:

```
MDISK 191 XXXX AUTOV 008 DSKPM8 MR
```

The example specifies an 8 cylinder minidisk to be allocated on volume DSKPM8. The installation EXTENT CONTROL file must contain the required entries.

```
MDISK 191 XXXX AUTOV 008 DSKPM8 MR READ1 WRITE2
```

The example specifies an 8-cylinder minidisk to be allocated on volume DSKPM8. The installation EXTENT CONTROL file must contain the required entries. This example also sets the read and write passwords to READ1 and WRITE2, respectively.

**Note:** For more examples of DirMaint automatic allocation methods, see [“AMDISK” on page 25](#).

9. The AMDISK and MDISK control statements can also take the traditional form in which you specify all parameters including *devtype*, a specific *volser*, and the starting cylinder or block number.
10. The AMDISK and MDISK control statements are stripped from the new directory entry when ADD processing begins. A batch request is generated for each MDISK statement and ADD processing continues. ADD processing will reorder the contents of the DIRECT file. The comments may no longer be placed as desired. Should the MDISK (AMDISK) request fail, the user will be notified of the error as the request is processed.

**Note:** The MDISK statements are removed from the new directory entry and passed directly to the DIRMAINT server for AMDISK command processing. A not valid syntax in the new directory entry will result in an error from the AMDISK command processor and the entire ADD request will fail.

The new directory entry you are adding can contain either the MDISK keyword or the AMDISK keyword. If the new entry contains the MDISK keyword, the command is passed to the DIRMAINT

server using the AMDISK 140A mode syntax. If the AMDISK keyword is used, the command is passed to the DIRMAINT server using the 150A mode syntax.

**Note:** The ADD command issuer will need to have the proper 140A and 150A cmdlevel authority or else the ADD request will fail.

The 150A command syntax provides a richer set of allocation request options than the 140A command syntax. Both have specific syntax requirements on the command that must be followed when entered into the directory entry being added. For instance, the AMDISK syntax for the 150A command set requires the use of the PWS keyword when providing passwords. So

```
AMDISK 191 XXXX AUTOV 008 DSKPM8 MR READ1 WRITE2
```

is syntactically invalid because the required keyword PWS is not provided. The correct syntax would be

```
AMDISK 191 XXXX AUTOV 008 DSKPM8 MR PWS READ1 WRITE2
```

However, if the inbound directory entry contained

```
MDISK 191 XXXX AUTOV 008 DSKPM8 MR READ1 WRITE2
```

DIRMAINT will consider this a 140A command level request and the command will process correctly.

11. LINK statements are handled through batch in the same way that MDISK statements are handled. They are removed from the new directory and added individually.
12. If the new entry contains statements in error, the entire ADD request is rejected.
13. The following directory entry names are restricted: \$DIRCTL\$, \$DIRGRP\$, SYSTEM, and ALL. Attempts to add a directory entry with one of these names is rejected.
14. If an INCLUDE statement in a new *userid* entry invokes a nonexistent profile, an error message is issued and the entry is rejected.
15. If an attempt is made to add a directory entry *LIKE* a stored prototype that does not exist on the DIRMAINT server, an error message is issued and the entry is rejected.
16. As the USER or IDENTITY statement within a stored prototype directory entry must be modified to reflect the user ID and password of the entry being added, it is possible that the addition of the user ID/password values to the statement could cause its length to exceed 71 characters. If this occurs the ADD request is rejected.
17. A way to add a user with a pre-defined configuration is to first create a *prototype* file under the name *fn PROTODIR*. Install it on any disk on the DIRMAINT server using the FILE command. For example:

```
DIRM FILE fn PROTODIR A
```

During installation, DirMaint installs two prototypes for typical CMS and LINUX virtual machines. The sample prototype files CMS SAMPDVH and LINUX SAMPDVH are copied as CMS PROTODIR and LINUX PROTODIR on the DirMaint server 1DF disk.

Note that the use of these prototypes requires that the EXTENT CONTROL file be updated with GROUP entries for CMSGROUP and LINGROUP.

As many unique PROTODIR files as needed can be installed in this manner. The user ID inside such files on the USER or IDENTITY statement must match the file name under which the prototype file is installed. All minidisks in the prototype definition must be allocated using automatic minidisk allocation. The new password must be specified when adding a user *LIKE* the prototype. An account number is optional.

If the prototype definition includes an ACCOUNT or a \*AC= record, each account number is checked by the ACCOUNT\_NUMBER\_VERIFICATION\_EXIT. If the ACCT *newvalue* option is specified on the command invocation, the *newvalue* must be contained in one of the following:

- The list of valid secondary account numbers on the ACCOUNT statement within the named PROTODIR file

- The list of valid tertiary account numbers on \*AC= records within the named PROTODIR file

If the ACCT *newvalue* is not contained in one of the above lists, the ACCOUNT\_NUMBER\_VERIFICATION\_EXIT must be enabled to treat the ADD-ACCT command as exempt from this requirement; otherwise, the ADD command will be rejected.

- An MDISK directory statement may be followed by DASDOPT or MINIOPT directory statements. An AMDISK statement may also be followed by DASDOPT or MINIOPT statements, if neither the BLKSIZE nor the LABEL keywords have been used on the AMDISK record. If either of the BLKSIZE or LABEL keywords have been used on the AMDISK record, the minidisk is assigned to the DATAMOVE machine(s) for formatting, and processing for a subsequent DASDOPT or MINIOPT will not find the associated parent MDISK statement in the user's directory entry, causing the ADD to fail, with message DVHMOP3302E.
- Duplicate comment records starting with the string \*DVHOPT are restricted. Any such lines will be removed from the supplied directory entry. \*DVHOPT records are maintained by DirMaint. On ADD and REPLACE commands, only the first \*DVHOPT record is maintained in the directory entry.
- Subconfiguration entries are used to specify definitions within a multiconfiguration user that are associated with a specific single system image cluster member. To create a multiconfiguration user, use the following steps:
  - Issue an ADD command to add a *userid* using an identity entry.
  - Issue an ADD command to add a *subconfig* using a subconfiguration entry. The BUILD option must be specified to link the subconfiguration entry to the user ID created in the step above.
- If the identity specified in the BUILD option does not exist, the ADD request will fail.
- If a BUILD statement for the node ID specified in the BUILD option already exists, the ADD request will fail.
- When the definition of the user ID being created contains STORAGE, MAXSTORAGE, or CLASS statements, their values will be moved to the corresponding fields in the USER or IDENTITY statement. For example, if you add new user ID JAMES with these statements:

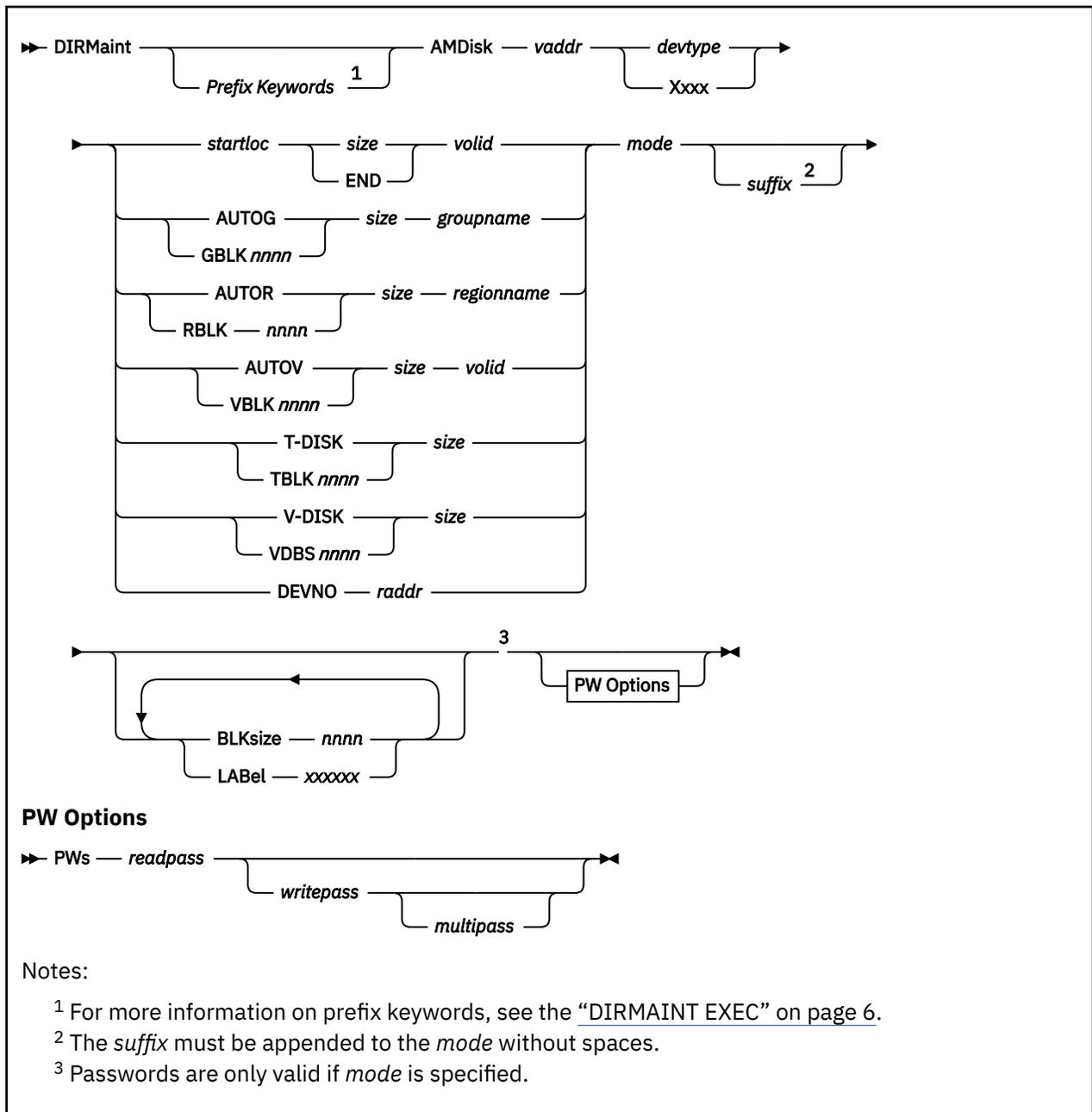
```
USER JAMES A1B2C3
CLASS G
STORAGE 8M
MAXSTORAGE 16M
```

then when you retrieve a copy of the directory entry for user JAMES it will contain the following:

```
USER JAMES A1B2C3 8M 16M G
```

- If a DASD management work unit associated with this command becomes stalled or hung on the DATAMOVE machine, manual intervention may be necessary in order for the work unit to complete. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.
- Password phrases are not supported for this command. If you specify the PW operand with a password phrase, command parsing errors may occur.
- If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

# AMDISK



## Purpose

Use the AMDISK operand of the DIRMAINT command to add a new minidisk to a user, identity, or subconfiguration entry.

## Operands

### vaddr

is the virtual device address of the new minidisk.

### devtype

is the device type of the volume to which the minidisk is assigned. Otherwise, if automatic allocation is used, specify the device type as Xxxx. The automatic allocation options include these operations:

- AUTOG and GBLK
- AUTOR and RBLK
- AUTOV and VBLK

The real device type will be taken from the REGIONS section in the EXTENT CONTROL file for that volser.

**startloc**

is a decimal cylinder relocation factor or a decimal FB-512 device block number that specifies the beginning of the minidisk extent corresponding to cylinder 0 or block 0 of the user's minidisk.

It is not advisable to define a minidisk starting at real cylinder 0, unless it is to be used by operating system (OS) indexed sequential access method (ISAM), in which case it must begin at real cylinder 0. If you do assign a minidisk beginning at real cylinder 0, the user who owns it must realize that the minidisk label is the real label that the user and the system use to identify the disk. Remember that CP-owned volumes must not have minidisks beginning at real cylinder 0. Generally, CKD DASD extents should begin with cylinder 1 or higher, and FBA DASD extents should begin with block 32 or higher.

**AUTOG**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a volume within a specified group. The group must have been defined in the EXTENT CONTROL file, and the regions within the group must also be defined.

**GBLKnnnn**

is functionally equivalent to AUTOG except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: GBLK512, GBLK0512, GBLK1024, GBLK2048, GBLK4096, GBLK1K, GBLK2K, and GBLK4K. In addition, GBLK800 or GBLK0800 may be specified for device types 3350, 3375, and 3380.

The disk will not be formatted for CMS use by DirMaint unless the BLKSIZE or LABEL operands are used, but the number of CMS blocks of a given size per cylinder is used to calculate the number of CKD cylinders or FB-512 blocks necessary to satisfy the request.

**AUTOR**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a specified region or volume. The region must have been defined in the EXTENT CONTROL file.

**RBLKnnnn**

is functionally equivalent to AUTOR except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: RBLK512, RBLK0512, RBLK1024, RBLK2048, RBLK4096, RBLK1K, RBLK2K, and RBLK4K. In addition, RBLK800 or RBLK0800 may be specified for device types 3350, 3375, and 3380.

The disk will not be formatted for CMS use by DirMaint unless the BLKSIZE or LABEL operands are used, but the number of CMS blocks of a given size per cylinder is used to calculate the number of CKD cylinders or FB-512 blocks necessary to satisfy the request.

**AUTOV**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a specified volume.

**VBLKnnnn**

is functionally equivalent to AUTOV except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: VBLK512, VBLK0512, VBLK1024, VBLK2048, VBLK4096, VBLK1K, VBLK2K, and VBLK4K. In addition, VBLK800 or VBLK0800 may be specified for device types 3350, 3375, and 3380.

The disk will not be formatted for CMS use by DirMaint unless the BLKSIZE or LABEL operands are used, but the number of CMS blocks of a given size per cylinder is used to calculate the number of CKD cylinders or FB-512 blocks necessary to satisfy the request.

**T-DISK**

provides temporary disk space to a user upon logging on. This space is made available from preallocated system disk space. The owner must initialize or format this minidisk at each logon. It will be a part of the owner's virtual configuration until logoff or disk detachment, at which time the owner loses all files and the space is returned for reallocation to another user. If concerned about security, the owner should initialize or reformat the minidisk before releasing it.

**TBLKnnnn**

is functionally equivalent to T-DISK except that the size is specified in units of CMS 512, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: TBLK512, TBLK0512, TBLK1024, TBLK2048, TBLK4096, TBLK1K, TBLK2K, and TBLK4K.

**V-DISK**

provides virtual disk space to a user upon logging on. The V-DISK may already exist when the user links to it, providing other users that previously linked to it are still linked when the user logs on. Otherwise, the owner must initialize or format this minidisk at each logon. It remains a part of the owner's virtual configuration until logoff or disk detachment. When the last linked user logs off or detaches the V-DISK, the file space is returned for reallocation to another user.

**VDBSnnnn**

is functionally equivalent to V-DISK except that the size is specified in units of CMS 512, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: VDBS512, VDBS0512, VDBS1024, VDBS2048, VDBS4096, VDBS1K, VDBS2K, and VDBS4K.

**size**

is a 1 to 8 digit decimal number specifying the size of the new minidisk in CKD cylinders or FBA blocks. For allocation requests involving block sizes, allocation will be converted to the exact number of FB-512 blocks for FBA devices or to the minimum number of cylinders needed to contain the specified number of CMS blocks for CKD devices.

**END**

specifies that the minidisk will extend from the assigned starting cylinder or block to the end (last cylinder or block) of the volume in which it is defined. This parameter is valid for fixed allocation only.

**valid**

specifies the volume label of the DASD volume, 1 to 6 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +, on which the minidisk is to be allocated, or the reserved identifier of &SYSRES. This applies to fixed allocation (where a specific starting cylinder or block number is specified) and to automatic allocation on a volume (AUTOV, VBLKnnnn).

**Note:** The reserved identifier of &SYSRES should not be used as the *valid* in an SSI-enabled directory where the SSI option is used on the DIRECTORY statement. This will result in a DIRECTXA error and failure of the AMDISK command.

**groupname**

specifies the name of the DASD allocation group from which DirMaint will pick the *volser* and starting CKD cylinder or FBA block. Groups consist of one or more specific *regions*. This is valid for AUTOG and GBLKnnnn requests only. Groups must be defined in the EXTENT CONTROL file, and can be 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +.

**regionname**

specifies the name associated with the *region*, 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +, on which the minidisk is to be allocated. This is valid for AUTOR and RBLKnnnn requests. The named region must be defined in the REGIONS section of the EXTENT CONTROL file. The region defines a segment on a specific DASD volume for allocation.

**DEVNO raddr**

specifies a full volume minidisk.

**raddr**

specifies the real device number of the full volume minidisk.

**mode**

specifies the user's mode of access at logon. If the minidisk owner allows other users access to the minidisk by establishing access passwords, then the access conditions may be modified by such links. Valid access modes are:

**Mode****Definition****R**

Read-only (R/O) access is desired. The access is not given if any other user has a link to the disk in write or exclusive status.

**RR**

Read-only (R/O) access is desired, even if another user has a link to the disk in write status. Access is denied if any user has an exclusive link.

**W**

Write access is desired. The disk is not accessible if any other user has a link to the disk in read or write status.

**WR**

Write access is desired, but read-only access is acceptable if any other user has a link to the disk in read or write status. Access is denied if any user has an exclusive link.

**M**

Multiple access is desired. A write link to the disk is given unless another user already has write or any exclusive access to it; in which case, the disk is not accessible.

**MR**

Write access to the minidisk is desired unless another user already has write or any exclusive access to it. In this case, read-only access to the minidisk is allowed. Access is denied if any user has an exclusive link.

**MW**

Unconditional write access to the disk is desired. Access is denied if any user has an exclusive or stable link.

**suffix**

specifies an optional mode suffix to be directly appended to the access mode. The suffix letters can be combined as follows (note that while CP requires they be specified in this order, DirMaint allows them in any order and will reorder them appropriately):

1. V or null, plus
2. S, E, or null (S and E are mutually exclusive), plus
3. D or null.

Therefore, the only valid combinations are: V, S, E, D, VS, VE, VD, VSD, VED, SD, or ED. This suffix is in turn concatenated with the mode, with no intervening blanks. For example, RV, RRS, WVE, and MVSD are all valid.

**Suffix****Definition****V**

Tells CP to use its virtual reserve/release support in the I/O operations for the minidisk.

**S**

Authorizes the virtual machine to use the LINK command stable access modes (SR, SW, SM) against the specified user's minidisk.

**E**

Authorizes the virtual machine to use the LINK command exclusive and stable access modes (ER, EW, SR, SW, SM) against the specified user's minidisk.

**D**

Tells CP that the device should not be defined when the virtual machine initially logs on or is autologged, but to defer doing so until an explicit LINK command is issued for that device.

**BLKsize *nnnn***

specifies the block size to use when formatting the new extent. The valid values include:

512  
1024  
2048  
4096  
1K  
2K  
4K

If the BLKSIZE parameter is not specified, but the LABEL parameter is specified, the disk will be formatted with the default blocksize determined by the CMS FORMAT command. If neither BLKSIZE nor LABEL are specified, the disk will not be formatted.

**LABEL *xxxxxx***

specifies the minidisk label to use when formatting the new extent. The labels are 1 to 6 alphanumeric characters.

If the LABEL parameter is not specified, but the BLKSSIZE parameter is specified, the disk will be formatted with a default label consisting of the minidisk address preceded by as much of the userid as will fit a 6 character label. If neither LABEL nor BLKSIZE are specified, the disk will not be formatted.

**PWs**

informs DirMaint that the following parameters are to be considered passwords.

***readpass***

is a password that allows a user other than the owner to get a read-only (R/O) link to the minidisk if the other user knows this 1 to 8 character password.

***writepass***

is a password that allows a user other than the owner to get a WRITE link to the minidisk if the other user knows this 1 to 8 character password.

***multipass***

is a password that allows a user other than the owner to get a MULTIPLE link to the minidisk if the other user knows this 1 to 8 character password.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Access mode and access passwords can be set and changed by the owner. If they are not entered with the AMDISK operand, the CP default will be access mode W, with no access passwords.
3. Due to the syntax of the CP MDISK statement, the use of access passwords requires that a mode be specified.
4. You may deliberately want to allocate a specific extent on a volume. If this extent would cross region boundaries or occupy a reserved area, you must have issued the EXTNCHK OFF operand before entering the AMDISK operand.
5. A profile directory entry may not be the target of the AMDISK operand.
6. A specification of END for extent size is only valid for permanent (not T-DISK) fixed-allocation minidisks.
7. When the *valid* used on the AMDISK command does not exist in the EXTENT CONTROL file the volume will be allocated explicitly. The devtype value provided (3390, 3380) will be used as the default device type. If the volume is attached to the system, DirMaint will query the device to determine the maximum size. In an SSI environment, the queries are issued on each SSI member and the maximum size of the query responses is used. When the volume is not attached, DirMaint will use the default maximum size that is found in the DEFAULTS section of the EXTENT CONTROL file or the

DEFAULTS DATADVH file (for example, a 3390 device will default to a maximum allocatable cylinder value of 1113).

8. The reserved identifier of &SYSRES should not be used as the *valid* in an SSI-enabled directory where the SSI option is used on the DIRECTORY statement. This will result in a DIRECTXA error and failure of the AMDISK command.
9. If a DASD management work unit associated with this command becomes stalled or hung on the DATAMOVE machine, manual intervention may be necessary in order for the work unit to complete. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.
10. If the AT prefix is specified for a subconfiguration entry when the AMDISK operand is issued, the AT keyword value will be used to select the DATAMOVE machine configured at the specified node. In this respect, the FORuser prefix keyword must specify the subconfiguration entry name in order to modify a subconfiguration entry with the AMDISK operand.
11. If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

### Examples

1. DIRM AMD 193 3390 15 4 VOL005

Adds a new 4-cylinder minidisk with a virtual device address of 193. It is to be allocated on a 3390 volume, *valid*= VOL005, at real cylinder number 15. The access mode will default to W (the CP default value), and no access passwords will be set.

Extent validation may fail and cause the transaction to fail if the specified extent would overlap an existing minidisk.

2. DIRM AMD 192 X AUTOV 10 VOLXYZ

This requests automatic allocation for a 192 minidisk of 10 cylinders. The allocation is to be made on volume *valid*= VOLXYZ.

3. DIRM AMD 192 X VBLK4096 1800 VOLABC

This requests automatic allocation of a minimum of 1800 CMS 4K blocks on VOLABC as defined in the EXTENT CONTROL file. Examples: if VOLABC is a 3390, 10 cylinders will be allocated; if 3380, 12 cylinders; if 3375, 19 cylinders (1824 blocks total); or, 14400 FB-512 blocks for an FBA device.

4. DIRM AMD 23F X AUTOG 8 MSSVOLS

This is a request for a new 23F minidisk of 8 cylinders. This is quite like the AUTOV example, except that the allocation is to be made automatically on a region within group MSSVOLS. The group name, MSSVOLS, and all volumes within it, must be currently defined in the EXTENT CONTROL file.

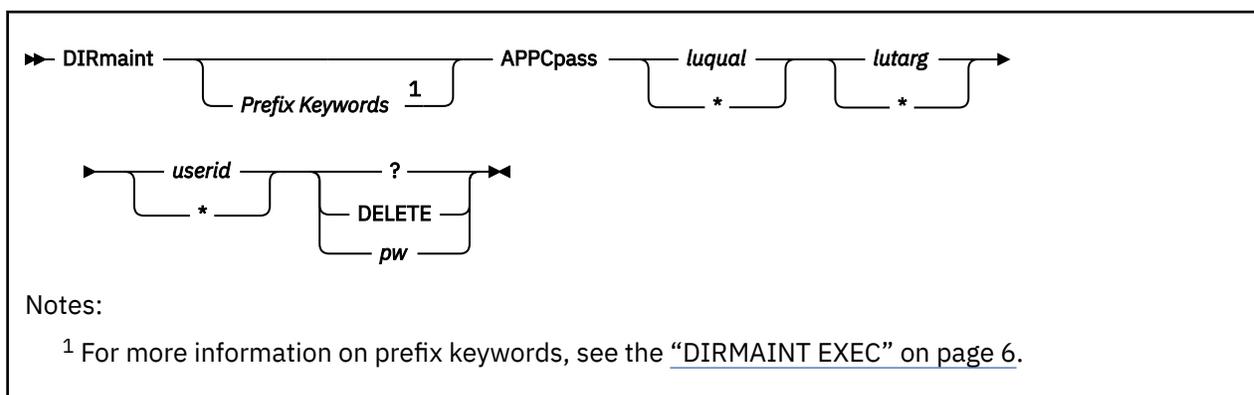
5. DIRM AMD 199 FB-512 T-DISK 480

Requests a temporary minidisk of 480 blocks, on a 9336 or other FBA device be made available as virtual device address 199 for each terminal session.

6. DIRM AMD 123 9345 DEVNO 839 MR PWS RPASS WPASS

This assigns the real 9345 DASD volume at address 839, whatever the volume serial identification may be, starting at cylinder 0 for the entire volume as virtual address 123. This is similar to having a DEDICATE 839 123, except that other users may link to this DEVNO minidisk by virtue of the passwords.

## APPCPASS



### Purpose

Use the APPCPASS operand of the DIRMAINT command to add a new APPCPASS statement to a profile, user, or identity entry. This operand can also be used to delete, query, and change these statements.

### Operands

#### *luqual*

represents the 1 to 8 alphanumeric character string that identifies whether:

- the target resource is located in a TSAF collection (in which case the values are \*IDENT or \*USERID).
- the target resource resides in the SNA network (in which case you must specify the communication server name).

The asterisk (\*) allows the APPCPASS operand to operate on all APPCPASS statements in the directory definition regardless of the LU qualifier.

#### *lutarg*

is the 1 to 8 byte character string that identifies the target LU name. If the LU qualifier is \*IDENT, the target LU must be 0 (zero).

The asterisk (\*) allows the APPCPASS operand to operate on all APPCPASS statements in the directory definition regardless of the LU target specified.

#### *userid*

is the 1 to 8 byte user ID of the target LU that authorizes the communication initiator to access resources located on the target LU.

The asterisk (\*) allows the APPCPASS operand to operate on all APPCPASS statements in the directory definition regardless of the user ID.

#### *?*

displays APPCPASS statements that meet the specified selection criteria.

#### **DELETE**

causes the APPCPASS statements that meet the specified selection criteria to be deleted from your directory definition.

#### *pw*

is the 1 to 8 byte password that corresponds to the password of the *userid* at the target LU.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Use of the PROMPT prefix operand will result in command link prompting for a missing password.
3. For information regarding the use and specification of APPCPASS statements, see [z/VM: CP Planning and Administration](#).
4. Because of the syntax of this operand, DELETE and ? cannot be used as passwords. If either of these is required as passwords, request that your administrator use DIRM GET and DIRM REPLACE.
5. The wild card character (\*) may be used in one or more positions to qualify a set of APPCPASS statements to be queried, DELETED, or have their passwords changed. A specific APPCPASS statement is qualified using the keyword and the next three tokens.
6. It is better from a security standpoint not to enter access passwords on the command line. If the password is omitted, and the system prompts you for it, then it can be entered in the *display inhibit* mode. This may inhibit the use of complex passwords. If you enter the password directly on the command line, you should follow installation guidelines for security.

## Directory Entry Samples

The following is a sample directory entry. Below the sample is a description of what would change in the entry due to several different invocations of the DIRM APPCPASS command operand.

The execution of each of these commands is contingent upon the users supplying the correct CP logon password for USERA when DirMaint prompts the user.

```

USER USERA LOGPASS 2M 2M G
ACCOUNT AAAAAAAAAA DIST
IPL CMS
APPCPASS LUQUAL1 LUTARG2 USERA PWA
APPCPASS LUQUAL3 LUTARG4 USERB PWB
APPCPASS LUQUAL3 LUTARG2 USERA PWA
APPCPASS LUQUAL3 LUTARG2 USERB PWB
CONSOLE 009 3215
SPOOL 00C 2540 R
SPOOL 00D 2540 P
SPOOL 00E 1403
LINK MAINT 190 190 RR
MDISK 191 3380 100 5 SYSRES M

```

Response messages containing the information on the last two APPCPASS statements in the sample directory entry will be sent to the USERA display if the following is issued by USERA:

```
DIRM APPC LUQUAL3 LUTARG2 * ?
```

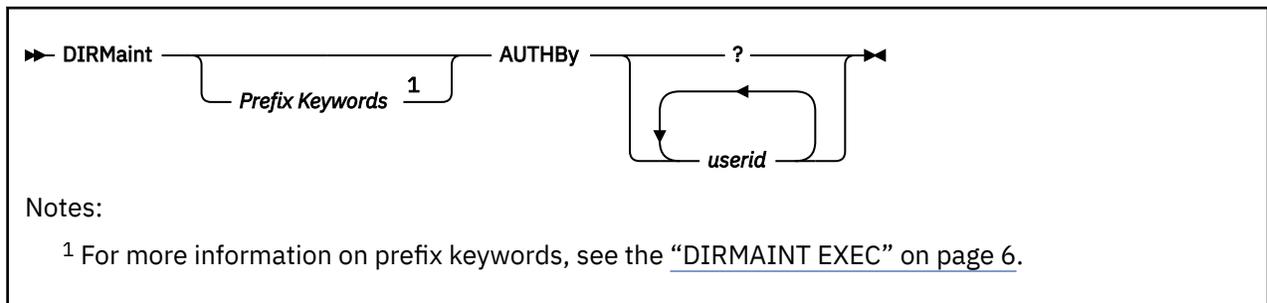
In the sample directory entry, the password on all the APPCPASS statements will be changed to PWC if the following is issued by USERA:

```
DIRM APPC * * * PWC
```

In the above directory entry, the last three APPCPASS statements will be deleted if the following command is issued by USERA.

```
DIRM APPC LUQUAL3 * * DELETE
```

## AUTHBY



### Purpose

Use the AUTHBY operand of the DIRMAINT command to update the AUTHBY CONTROL file to allow the listed user ID to use the BYUSER prefix.

### Operands

?

requests that the users currently authorized to use the BYUSER prefix be shown.

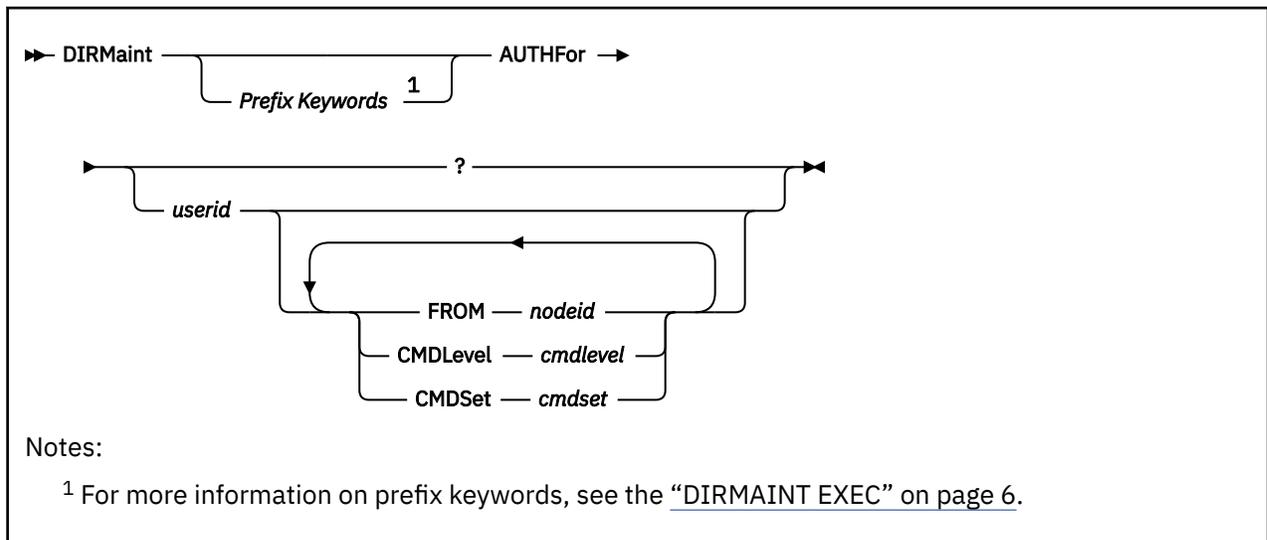
*userid*

is a list of one or more user IDs that should be added to the current AUTHBY CONTROL file.

### Usage Notes

You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.

## AUTHFOR



### Purpose

Use the AUTHFOR operand of the DIRMAINT command to authorize one user to act on behalf of another user.

### Operands

?

requests that the current users authorized to act on behalf of the command issuer be reported.

#### **userid**

specifies the user ID that is being authorized to act on behalf of the command issuer.

#### **FROM nodeid**

specifies the node ID of the user that is being authorized to act on behalf of the command issuer.

#### **CMDLevel cmdlevel**

specifies the command level being authorized.

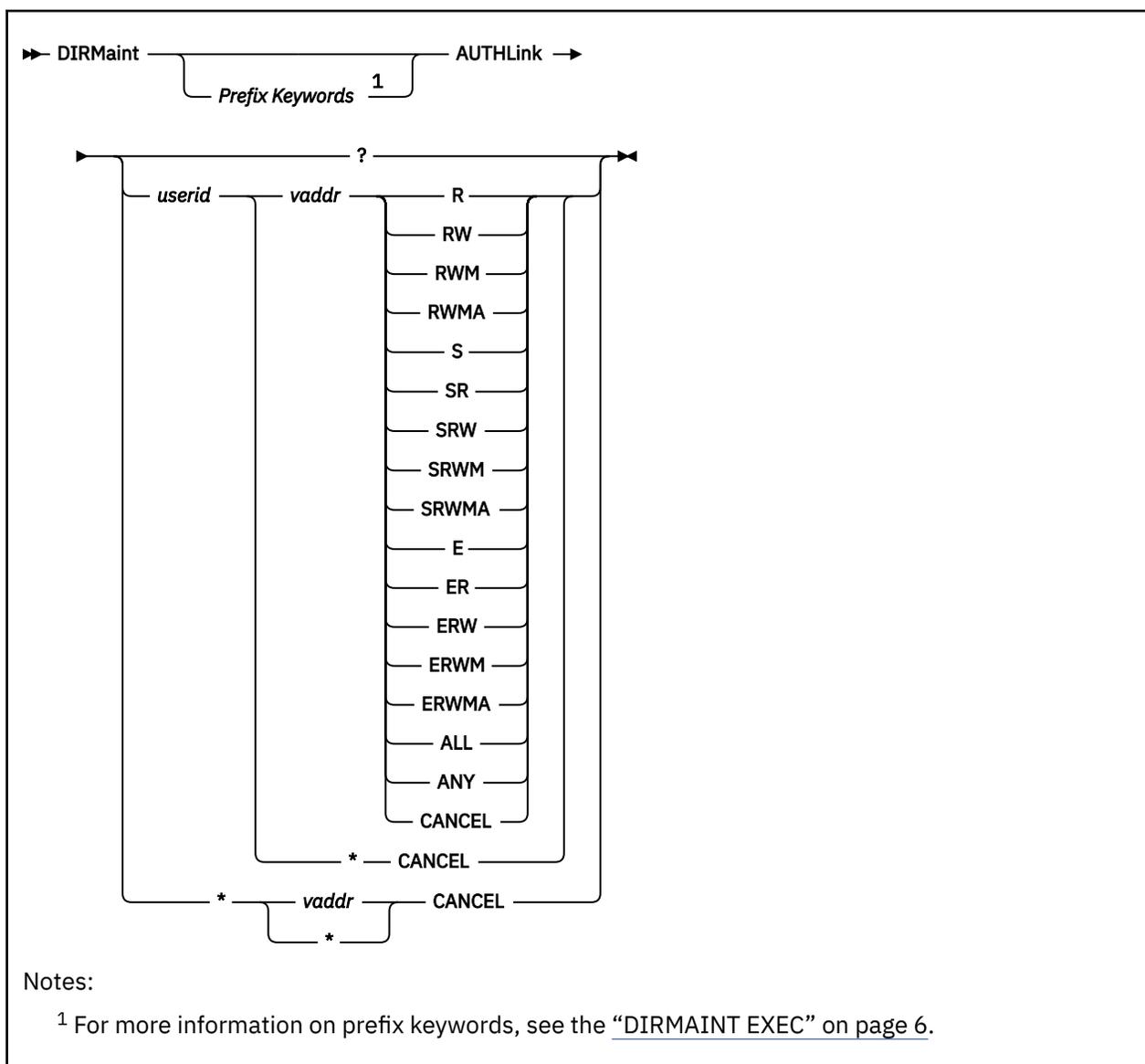
#### **CMDSet cmdset**

specifies the command set being authorized.

### Usage Notes

You may use one or more of the prefix keywords (REQuest, TOSys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.

## AUTHLINK



### Purpose

Use the AUTHLINK operand of the DIRMAINT command to allow a specified user or users to issue DIRM LINK commands without having to know the minidisk password for the disk being linked.

### Operands

?

requests that the current link authorizations be reported.

*userid*

\*

specifies the user being granted the authorization. When used in place of the *userid* operand, the \* value indicates that all previously granted authorities to link to the specified disk are to be cancelled. The \* is used with the CANCEL keyword exclusively.

***vaddr*****\***

specifies the address being authorized or cancelled. When used in place of the *vaddr* operand, the \* value indicates that the specified user's authority to link to any of the owners disks is to be cancelled. The \* is used with the CANCEL keyword exclusively.

**R**

requests that Read links be authorized.

**RW**

requests that Read and Write links be authorized.

**RWM**

requests that Read, Write, and Multi links be authorized, excluding MW.

**RWMA**

requests that Read, Write, and Multi links be authorized, including MW.

**S**

requests that Stable links be authorized.

**SR**

requests that Stable Read links be authorized.

**SRW**

requests that Stable Read/Write links be authorized.

**SRWM**

requests that Read, Write, and Multi links be authorized, excluding MW.

**SRWMA**

requests that Read, Write, and Multi links be authorized, including MW.

**E**

requests that Exclusive/Stable links be authorized.

**ER**

requests that Exclusive/Stable Read links be authorized.

**ERW**

requests that Exclusive/Stable Read/Multi links be authorized.

**ERWM**

requests that Read, Write, and Multi links be authorized, excluding MW.

**ERWMA**

requests that Read, Write, and Multi links be authorized, including MW.

**ALL**

allows any of the above modes to be specified.

**ANY**

allows any of the above modes to be specified.

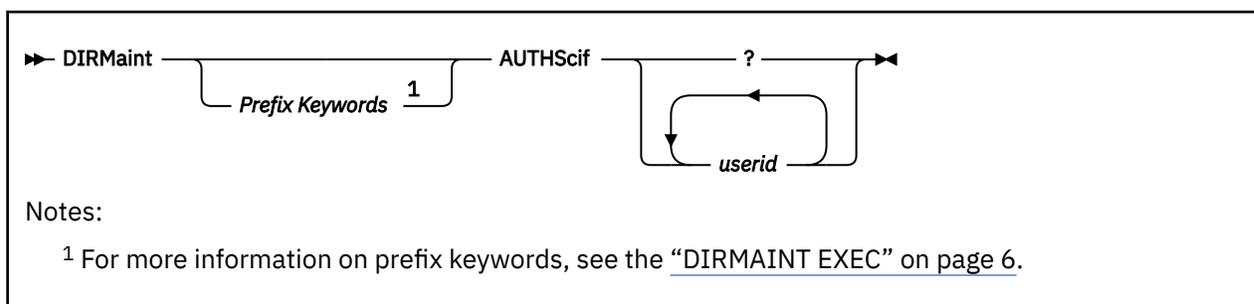
**CANCEL**

requests that the current link authorizations for the specified user/device combination be cancelled.

**Usage Notes**

You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.

## AUTHSCIF



### Purpose

Use the AUTHSCIF operand of the DIRMAINT command to allow a general user to authorize user IDs to act as secondary user IDs. The AUTHSCIF operand updates the AUTHSCIF CONTROL file, which controls users allowed to use the DIRMAINT SECUSER command.

### Operands

?

returns in a message a list of the user IDs that are authorized to use your user ID as a secondary user ID.

*userid*

is a list of one or more user IDs to be authorized to use your user ID as a secondary user ID.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. The use of SECUSER, AUTHSCIF, DSECUSER and DROPSCIF commands are closely related. The *userid* must have granted the issuing user authority through the AUTHSCIF command to use them as a secondary console. An error message will result if this has not been done. For example, if the *userid* WATCHER chooses to become the secondary console of the disconnected user SERVER, this sequence of commands is needed:

- a. WATCHER issues DIRM AUTHSCIF SERVER
- b. SERVER issues DIRM SECUSER WATCHER

When the secondary user facility is no longer needed, one of two commands would be needed:

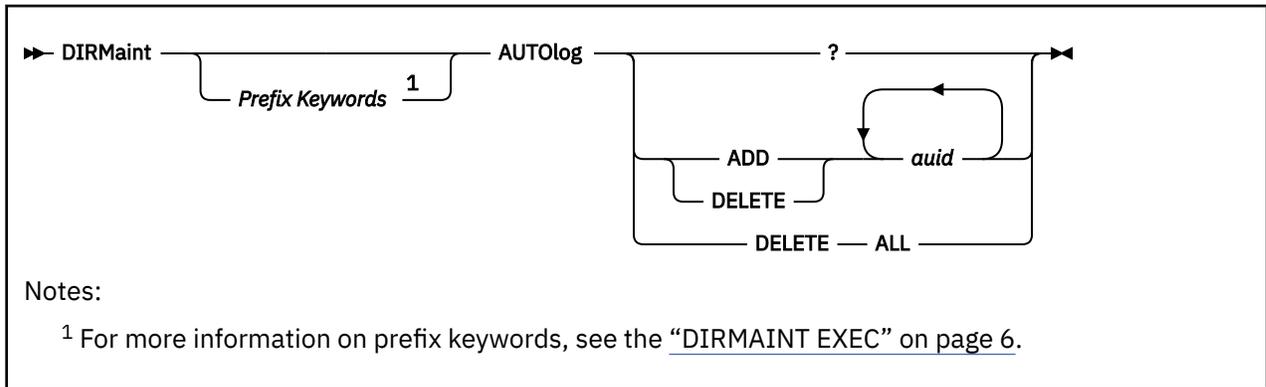
- a. WATCHER issues DIRM DSECUSER SERVER
- b. SERVER issues DIRM SECUSER OFF

If SERVER is not going to use WATCHER as a SCIF user again, the general authority can be cancelled if WATCHER issues:

```
DIRM DROPSCIF SERVER
```

The design allows the SCIF target *userid* (WATCHER) to have overall control of the usage of SCIF, in case of SERVER produced SCIF output which the target was not able to handle. SERVER has sole control of the time it is turned on, but WATCHER must authorize it first. If the SECUSER command is issued before the nominated *userid* has granted authority with the AUTHSCIF command, an error message will result.

## AUTOLOG



### Purpose

Use the AUTOLOG operand of the DIRMAINT command to add or delete a virtual machine name from an existing AUTOLOG or XAUTOLOG statement for a specified profile, user, or identity entry.

### Operands

?

specifies that the existing AUTOLOG or XAUTOLOG statement is to be queried.

#### ADD

specifies that a user ID (*auid*) is to be added to an existing AUTOLOG or XAUTOLOG statement or a new statement is to be created.

#### DELETE

#### DELETE ALL

specifies that a user ID (*auid*) is to be deleted from the AUTOLOG or XAUTOLOG statement. If the ALL keyword is used the entire statement is deleted.

#### *auid*

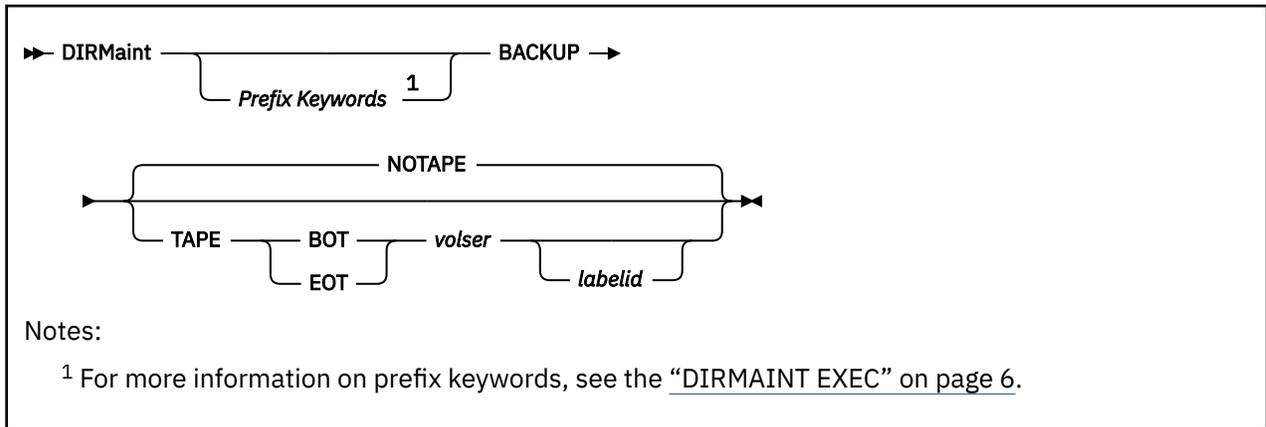
specifies a 1 to 8 character name of the virtual machine authorized to AUTOLOG or XAUTOLOG the specified user ID.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the AUTOLOG directory statement, see *z/VM: CP Planning and Administration*.
3. XAUTOLOG and AUTOLOG are synonymous but XAUTOLOG is the preferred directory keyword. Therefore, if the keyword AUTOLOG is encountered it is translated to XAUTOLOG.
4. A "DIRM XAUTOLOG ADD *auid*" may result in the creation of an XAUTOLOG statement, if there are no existing XAUTOLOG statements in the directory entry, or no more room on an existing XAUTOLOG statement. Similarly, a "DIRM XAUTOLOG DELETE *auid*" may result in the deletion of an XAUTOLOG statement if there are no more *auids* on the existing XAUTOLOG statement.
5. Specifying DELETE for a user ID not on an AUTOLOG or XAUTOLOG statement will generate an error message. Specifying ADD for a user ID already on an AUTOLOG or XAUTOLOG statement will also generate an error message.
6. A maximum of eight user IDs may be specified through AUTOLOG or XAUTOLOG.

7. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## BACKUP



### Purpose

Use the BACKUP operand of the DIRMAINT command to allow an authorized user to manually trigger the normally automatic backup procedure due to an unusual or emergency condition.

### Operands

#### NOTAPE

requests that a backup be done to the DirMaint primary backup disk. A monolithic copy of the current directory and all control files are placed on to this disk.

#### TAPE

requests that a backup be done to disk and then to a tape device.

#### BOT

#### EOT

controls the positioning of the tape device when tape backups are being made. BOT specifies that the backup will be made at the Beginning Of the Tape (if *labelid* is also specified, the backup will be made after the tape label). EOT specifies that the backup will be made at the End Of the Tape.

#### *volser*

specifies the volume serial of the tape, to be displayed in the message to the operator.

#### *labelid*

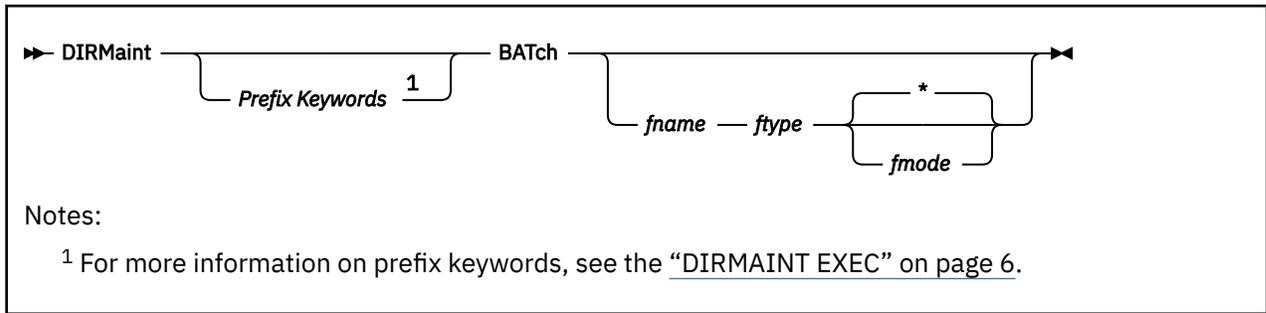
specifies the tape label, for verification that the correct tape has been mounted.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The backup procedure is executed automatically each day at the time specified in the DIRMAINT DATADVH file. If the system is down at the time specified in the DIRMAINT DATADVH file, backup will be automatically executed during DirMaint IPL.
4. The difference between the BACKUP operand and the USER BACKUP operand is that the USER BACKUP operand causes regeneration of the USER BACKUP file only. The BACKUP operand invokes the full backup procedure that backs up the auxiliary control files as well.

5. Use of BACKUP TAPE requires the BACKUP\_TAPE\_MOUNT\_EXIT be defined in the CONFIG\* DATADVH file(s). The IBM supplied sample (DVHXTAPE EXEC) uses simple messages to the operator. The exit may be customized to interface with a variety of tape library management programs.

## BATCH



### Purpose

Use the BATCH operand of the DIRMAINT command to allow users to submit any number of commands to the DIRMAINT service machine in one file. The commands will execute in a background work queue, as opposed to the interactive work queue, from which individual user commands are executed. DIRMAINT queries the background queue for work only when the interactive work queue is empty.

### Operands

#### *fname*

specifies the file name of the batch file.

#### *ftype*

specifies the file type of the batch file.

#### *fmode*

\*

specifies the file mode of the batch file.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. After entering DIRM BATCH (without operands), the user is placed in an XEDIT session, from which the user can enter any number of DIRMAINT commands, one per line. When command input is complete, the file is sent to DIRMAINT for processing. DIRMAINT sends messages when it opens the batch to begin work and when work on each command is begun and finished. If the submitting user ID is not logged on at that time, DIRMAINT logs these messages in the usual LOG file for that user.
4. The commands entered into the batch file while in the XEDIT session should not begin with the normal DIRM or DIRMAINT keyword as it does when a non-batch command is being issued. For example, if the user wishes to issue the command:

```
DIRM LINK USERA 111 222 DELETE
```

as one of the commands in a batch, the record in the batch file should be:

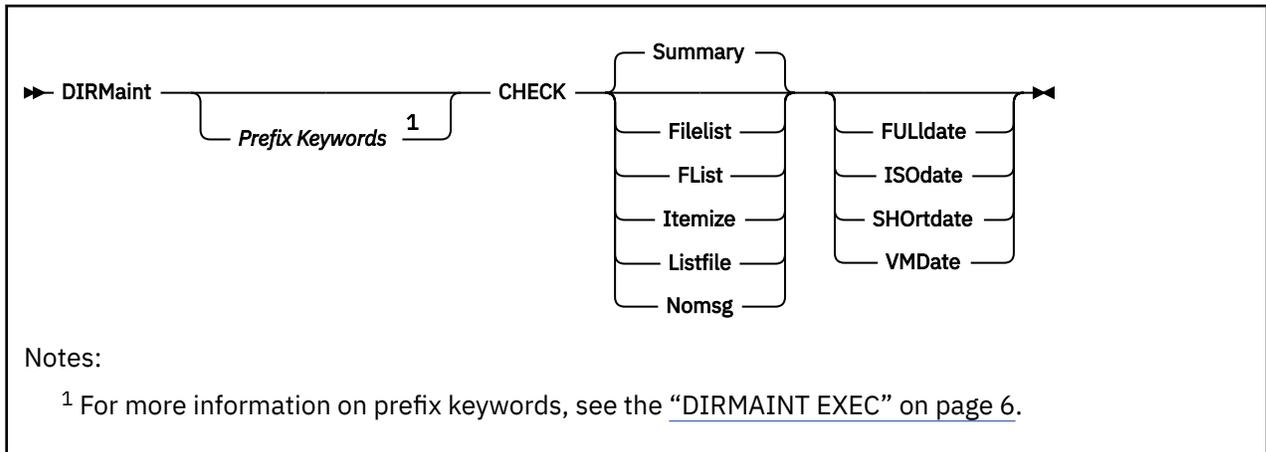
```
LINK USERA 111 222 DELETE
```

not

```
DIRM LINK USERA 111 222 DELETE
```

5. When the resulting file from a BATCH command is submitted for processing, it is also saved on the invoker's A-minidisk. The file is retrieved the next time the DIRM BATCH command is entered.
6. Transactions such as commands and operands entered into the batch file can not involve the transfer of any bulk data files. Operands such as ADD (unless the LIKE prototype facility is being used), MAIL, BATCH, FILE, PWMON LOCKOUT, PWMON REPLACE CONTROL, and REPLACE are examples of such transactions.
7. To enter commands that are longer than your screen width, you can scroll the edit session to the right as needed. Examples of long commands include DIRM DASD, LOADDEV SCPDATA, POSIXINFO, POSIXFSROOT, POSIXIUPGM, and POSIXIWDIR.

## CHECK



### Purpose

Use the CHECK operand of the DIRMAINT command to determine if there are extraneous DirMaint product files on any accessed disk or SFS directory.

### Operands

#### Filelist

Replaces the ITEMIZE diagnostic messages (DVH1371) with an actual EXEC FILELIST command to enable further manual checking for potential problems.

#### FList

Replaces the ITEMIZE diagnostic messages (DVH1371) with an FLIST command to enable further manual checking for potential problems.

#### Itemize

Produces a detailed diagnostic message (DVH1371) for each extraneous file found in your search order.

#### Listfile

Replaces the ITEMIZE diagnostic messages (DVH1371) with a LISTFILE command to provide further diagnostic information on potential problems. This information includes the date and time stamps and the disk label for each extraneous file found in your search order.

**Note:** The output will reside in DVHUCHK OUTPUT A.

#### Nomsg

Suppresses all diagnostic messages. The only remaining indication of any potential problems is a non-zero return code.

#### Summary

Suppresses the itemized diagnostic messages (DVH1371) but produces a summary diagnostic message (DVH1372) showing the total number of extraneous files found.

#### FULLdate

Uses the FULLDATE option on the FILELIST or LISTFILE commands.

#### ISOdate

Uses the ISODATE option on the FILELIST or LISTFILE commands.

#### SHORtdate

Uses the SHORTDATE option on the FILELIST or LISTFILE commands.

#### VMDate

Uses the VMDATE option on the FILELIST commands.

## Usage Notes

1. DirMaint prefix operands do not apply to this command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The CHECK command is completely handled within the virtual machine of the command issuer. No transaction is sent to the DIRMAINT service machine, therefore no password is required or requested for authentication, even if you have NEEDPASS set to YES.
4. The CHECK command is not valid within a BATCH command file.
5. The return code from the CHECK command is the number of extraneous files found.
6. This command will create a scratch file called DVHUCHK OUTPUT A. Except for the LISTFILE option, this scratch file will be erased when the command is complete. The FLIST option will also create and erase the CMS EXEC A file.
7. This command is used to investigate the cause of problems. Since duplicate files don't cause problems to the DIRMAINT machine, extraneous or duplicate files are not reported.
8. For more information about the FULLDATE, ISODATE, SHORTDATE, or VMDATE operands, type HELP CMS FILELIST or HELP CMS LISTFILE or see [z/VM: CMS Commands and Utilities Reference](#).

## CHKSUM



Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the CHKSUM operand of the DIRMAINT command to verify the integrity of the source directory.

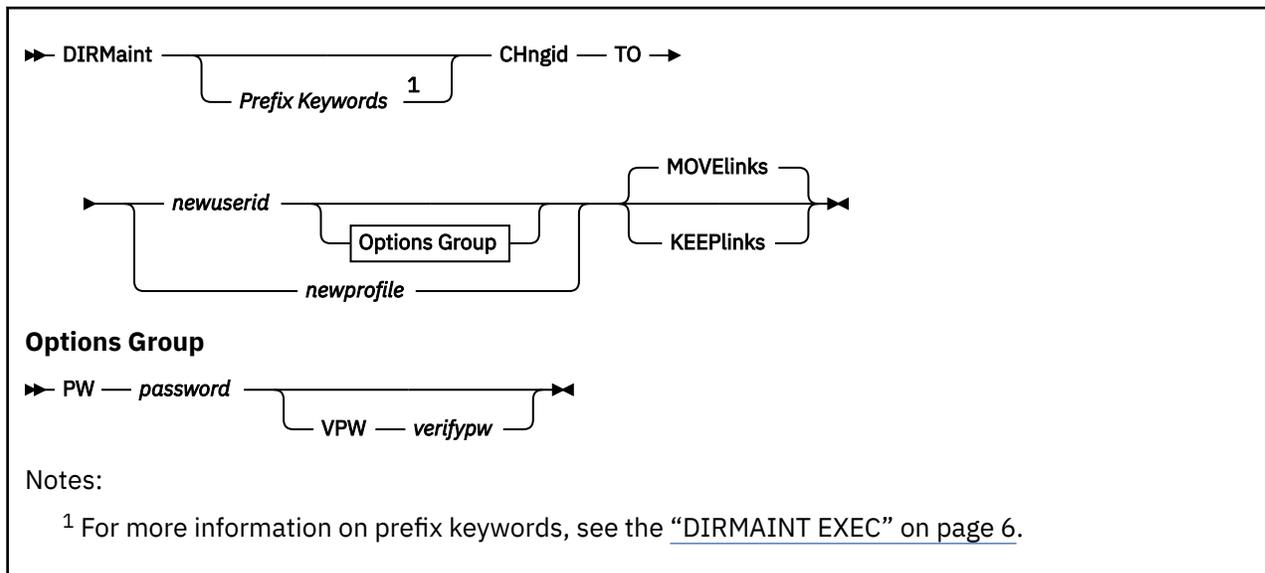
### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The act of generating or regenerating the directory initializes the checksum values. Once the directory source is initialized, it contains individual checksums for each directory entry. A checksum is a logical arithmetic sum of the contents of a directory entry. They are used to guard against partial loss of data due to DASD I/O errors. Checksums are recomputed each time a directory entry is altered, and compared against the stored value. If a mismatch is detected, your system administrator is informed and should take corrective action. These checksums are stored in a special record that follows each directory entry definition. These checksums are maintained by DirMaint as each transaction modifies the source directory.
3. Once the directory has been initialized, it should not be edited directly. Direct editing invariably introduces checksum errors, possibly for every entry if default serialization is allowed to take place. An editor may only be used as part of the cycle of transactions (GET operand, edit, REPLACE operand), when applied to a single directory entry at a time.

To make multiple updates to the entire directory (updates that may affect multiple users) see the "Making Multiple Updates to a Directory" appendix in the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*.

4. The CHKSUM operand reads through the source directory, generating a new checksum for each directory entry and comparing the newly computed value with the stored value. Checksum errors will be reported but not corrected.
5. If the number of definitions with faulty checksums is small, they can be corrected by performing a GET/Edit/REPLACE cycle on each definition in error. REPLACE processing will automatically regenerate the checksum value for each definition being replaced in the source directory. If there are a large number of definitions in error, consider initializing the directory from a monolithic copy (USER INPUT). New checksum values will be automatically recalculated for all definitions.

## CHNGID



### Purpose

Use the CHNGID operand of the DIRMAINT command to change the name of an existing profile or the user ID of an existing user or identity. The CHNGID operand cannot be used to change the name associated with a subconfiguration entry..

### Operands

#### TO

is a required keyword.

#### *newuserid*

specifies the new user ID name.

#### *newprofile*

is the new profile entry name.

#### **PW** *password*

specifies a new CP logon password that will be assigned to *newuserid*.

See Usage Note “10” on page 48.

#### **VPW** *verifypw*

specifies the same password again for typographical verification

#### **KEEPlinks**

requests that the links the devices owned by this user ID not be changed to reflect the new name.

#### **MOVElinks**

requests that the links the devices owned by this user ID be changed to reflect the new name.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Although CP allows a profile entry to be defined in the directory with the same name as a user ID or subconfiguration entry, DirMaint requires that every directory entry (either user ID, subconfiguration,

or profile) be unique from all others. Therefore, all directory entries in the CP directory managed by DirMaint must have a unique name/ID in respect to all other CP directory entries.

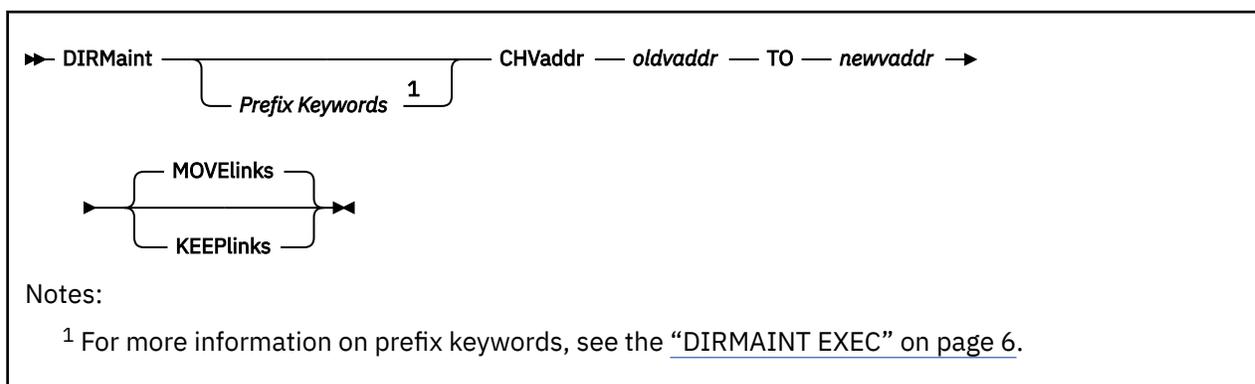
3. This operand is equivalent to the following series of operations:
  - a. GET (with the NOLOCK option) the existing user ID or profile. The existing user ID or profile DIRECT file will be sent to your reader.
 

**Note:** You must use the NOLOCK option when performing this step to avoid a PURGE failure problem when performing step “3.c” on page 48.
  - b. Read in the existing user ID or profile DIRECT file. Use the editor to change the user ID in the USER or IDENTITY control statement to *newuserid* or the profile name in the PROFILE control statement to *newprofile*. Rename the file to *newuserid* DIRECT or *newprofile* DIRECT.
  - c. PURGE the old user ID or profile with the NOCLEAN option.
  - d. ADD *newuserid* or *newprofile*.
4. You cannot change the name of a profile if any INCLUDE statements invoke it. If you try, an error message will be issued and the change will be rejected. User entries using the subject profile will be indicated in the returned messages. If you want to use CHNGID to change a profile name that is the target of INCLUDE statements, you must perform the following steps in order:
  - a. Use GET, edit, and REPLACE to delete all the INCLUDE statements that invoke the profile in question, or to change them so that they invoke another profile. An alternative would be to use the DIRM INCLUDE command. This command allows the deletion of INCLUDE statements.
  - b. Use CHNGID to change the profile name.
  - c. Use GET, edit, and REPLACE to replace the INCLUDE statements or to change them so that they invoke the profile by its new name. An alternative would be to use the DIRM INCLUDE command. This command allows the creation and alteration of INCLUDE statements.

It is easier to change a profile name this way:

- a. Use ADD to add a copy of the profile with the new name.
  - b. Use GET, edit, and REPLACE to change all the INCLUDE statements that invoke the profile in question so that they invoke the profile by its new name. An alternative would be to use the DIRM INCLUDE command. This command allows the creation and alteration of INCLUDE statements.
  - c. Use PURGE to delete the old copy of the profile.
5. The CHNGID operand can also change the user's logon password.
  6. Attempts to specify a password when changing a PROFILE result in the transaction being rejected.
  7. If an account record exists in the entry to be changed, the account number verification exit (ACCOUNT\_NUMBER\_VERIFICATION\_EXIT= in the configuration file) is called. The exit may reject the account numbers. If this is done the transaction is rejected.
  8. CHNGID processing will invoke the logon password user exit facility (if one is in use at the installation), even if there is no change in the logon password. This is done to advise the exit of the change in the user ID name. The rationale for this action is that an installation may be using the exit to maintain installation-defined files. If a password change is involved, the exit may subject the proposed password to the normal installation-defined checks.
  9. Any links to this device are changed to match the *newuserid* and *newprofile*. This is done as a batch request. Because the execution of such work may be deferred, it is possible that one or more of the linking user IDs might manually change their own LINK statements (using the LINK operand) before the low priority work is executed. If this occurs, the low priority command to change the link will fail when selected for execution. This error condition can be safely ignored.
  10. Password phrases are not supported for this command. If you specify the PW operand with a password phrase, command parsing errors may occur.
  11. If your system specifically has RACF installed as the ESM, see the [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## CHVADDR



### Purpose

Use the CHVADDR operand of the DIRMAINT command to change the virtual address of an existing device.

### Operands

#### *oldvaddr*

is the current virtual device address.

#### TO

is a required keyword.

#### *newvaddr*

is the virtual device address that the existing device address is to be changed to.

#### KEEPlinks

requests that links to this device, if any, not be changed to reflect the new virtual address.

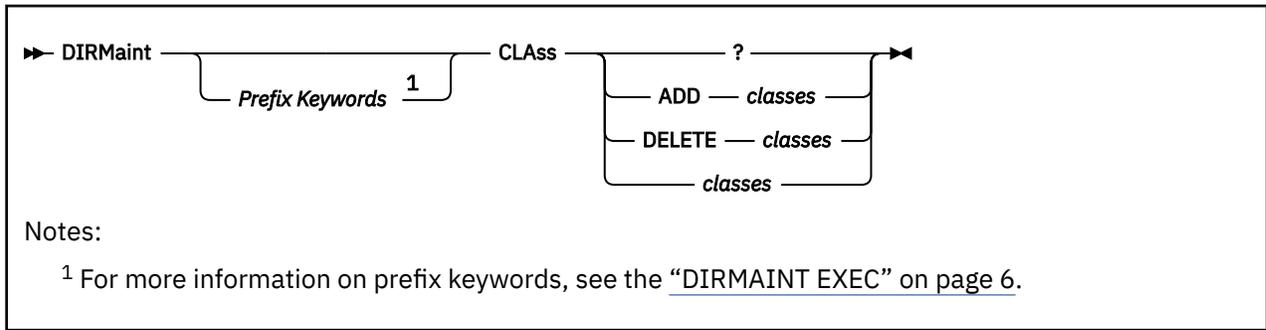
#### MOVElinks

requests that links to this device, if any, be changed to reflect the new virtual address.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. The control statement in the directory entry for user ID is updated to reflect the new *vaddr*.
3. If the device statement is either a LINK or an MDISK, any links to this device are changed to match the new *vaddr*. This is done as a batch request. Because the execution of such work may be deferred, it is possible that one or more of the linking user IDs might manually change their own LINK statements (using the LINK operand) before the low priority work is executed. If this occurs, the low priority command to change the link will fail when selected for execution. This error condition can be safely ignored.
4. If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## CLASS



### Purpose

Use the CLASS operand of the DIRMAINT command to enable you to add, delete, replace, or query the current class value for a profile, user, or identity entry.

### Operands

?

allows you to query your current class values.

#### **ADD classes**

#### **DELETE classes**

specifies the new class values to be added to the existing class values, or the class values that are to be deleted from the existing class values.

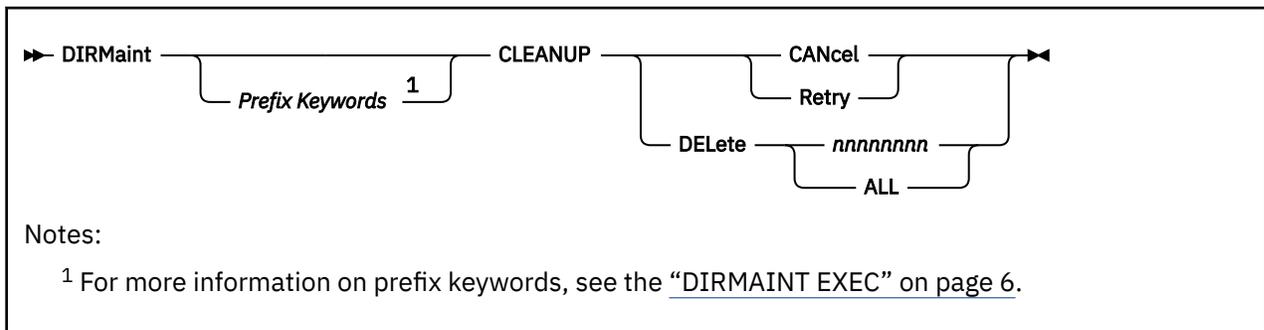
#### **classes**

when specified without the ADD or DELETE keyword, the new class values replace the current class information in the directory entry.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For a complete description of the CLASS directory statement and a discussion on how it relates to the USER or IDENTITY statement, see [z/VM: CP Planning and Administration](#).
3. An attempt to add a class value when it already exists on the USER or IDENTITY statement will be rejected.
4. An attempt to delete a class value from the USER or IDENTITY statement when that value does not exist will be rejected.
5. The class value specified with this command will be added to the corresponding field in the USER or IDENTITY statement.

## CLEANUP



### Purpose

Use the CLEANUP operand of the DIRMaint command to unhang a DATAMOVE machine by cleaning up internal files and retrying or canceling all workunits assigned to the machine. Also, you can subsequently use this operand to delete those debug files that DirMaint creates during cleanup processing.

### Operands

#### CANcel

unhangs the specified DATAMOVE machine and cancels all pending work units assigned to the DATAMOVE. This operand sends a cleanup transaction to the DATAMOVE machine. The cleanup completes when the DATAMOVE machine responds with the correct pending count and WORKUNIT CANCEL commands for all pending work units. All work units backed up to the hung DATAMOVE machine will be cancelled.

#### Retry

unhangs the specified DATAMOVE machine and retries all pending work units assigned to the DATAMOVE. This operand sends a cleanup transaction to the DATAMOVE machine. The cleanup completes when the DATAMOVE machine responds with the correct pending count and WORKUNIT RETRY commands for all pending work units. All work units backed up to the hung DATAMOVE machine will be retried.

#### DELETE nnnnnnnn

deletes any debug files saved during CLEANUP processing that are associated with the specified failed work unit. The work unit associated with a specific CLEANUP command is returned in message DVH3630I when the original CLEANUP command is entered.

**Note:** The CLEANUP debug files are the following system files as they were before the CLEANUP command was issued and copied, with the same file name and a file type of the work unit number:

- DATAMOVE CONTROL
- DVHLOCK DATADVH
- UNASSIGN DVHQUEUE
- PURGES PENDING
- DVHDMCTL QUEUE

Note that the actual failed work unit file with a file name of the work unit number and a file type of WUCFFAIL will not be deleted with this operand.

#### DELETE ALL

deletes all debug files saved during the processing of previous CLEANUP commands.

**Note:** The CLEANUP debug files are the following system files as they were before the CLEANUP commands were issued and copied, with the same file name and a file type of the work unit number associated with one of the specific CLEANUP commands:

## CLEANUP

- DATAMOVE CONTROL
- DVHLOCK DATADVH
- UNASSIGN DVHQUEUE
- PURGES PENDING
- DVHDMCTL QUEUE

Note that the actual failed work unit files with file names of the work unit numbers and a file type of WUCFFAIL will not be deleted with this operand.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.

Note that the FORuser keyword is used to identify a specific DATAMOVE machine. If FORuser is not specified, the default is 'DATAMOVE'.

2. When attempting to clean up a DATAMOVE machine, the following information may be used to determine the status of the DATAMOVE machine and the individual work units:
  - To see the state of the DATAMOVE machine, look for either the "Status:" label in the DIRMaint Query Datamove command output or the "Activity:" label in the DIRMaint STATUS Datamove command output. Valid states are ACTIVE, INACTIVE, and QUIESCED. If the DATAMOVE machine is ACTIVE, you can find the active work unit ID by looking for either the "WUCF:" label in the DIRMaint Query Datamove command output or the "CurUnit:" label in the DIRMaint STATUS Datamove command output. Other information from the DATAMOVE control file is also provided.
  - Current DirMaint work units may be queried using the DIRMaint STATUS Workunit All command. For information on individual work units, use either the DIRMaint STATUS Workunit or the DIRMaint Query Workunit command with a specific work unit ID to return data from the work unit control file.
  - When all configured DATAMOVE machines are unable to receive more work, new work units are placed on the DirMaint unassigned queue. These work units may be queried using the DIRMaint Query Unassigned command.

**Note:** Refer to the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on the DATAMOVE control file, work unit control files and the unassigned queue.

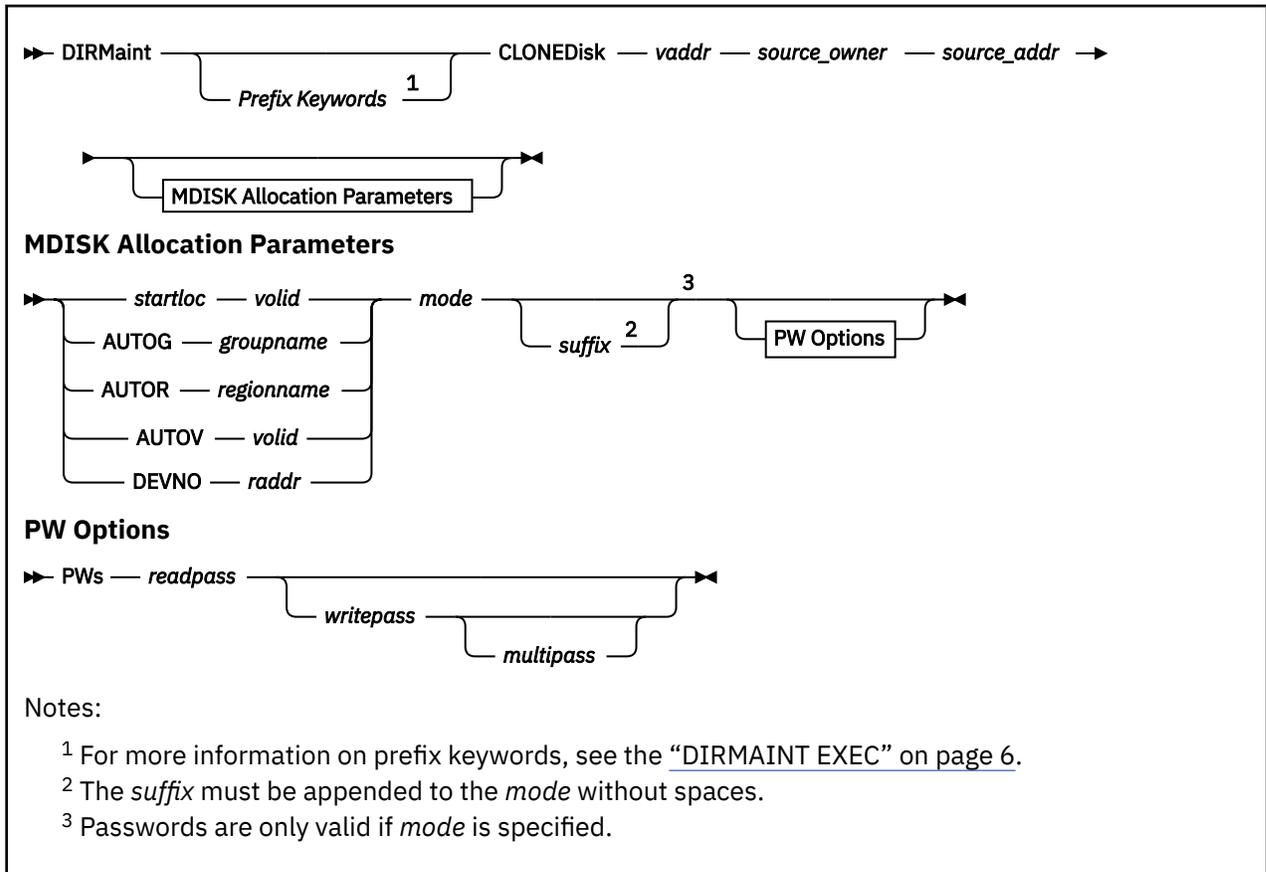
3. When the state of a DATAMOVE machine is QUIESCED, a CLEANUP retry may be used to reset the machine state and to retry all pending work units for that associated DATAMOVE machine. If the DATAMOVE machine returns to a QUIESCED state processing the work units, the work units may need to be cancelled in order to unhang the DATAMOVE machine. In this situation, use the DIRMaint CLEANUP CANCEL command to cancel all work units associated with the DATAMOVE machine. If the DATAMOVE machine does *not* return to a QUIESCED state but work units remain pending on the DATAMOVE machine, individual work units may be cancelled or retried using the DIRMaint WORKUNIT command.
4. When the CLEANUP operand is used to reset the state of a DATAMOVE machine that has been QUIESCED due to reaching the retry queue limit (configured using the DM\_MAXIMUM\_RETRIES statement), the cleanup processing will temporarily raise the retry queue limit to the current retry queue limit plus the configured unassigned queue limit, which is configured using the MAXIMUM\_UNASSIGNED\_WORKUNITS statement. This is done in order to allow work units which have backed up on the unassigned queue to be processed. If the DATAMOVE machine returns to a QUIESCED state handling the unassigned queue work units, additional DIRMaint CLEANUP commands may be needed to process the work units. But before issuing additional CLEANUP commands, the DirMaint DASD management subsystem should be disabled by loading an override configuration file (using the DIRMaint RLDDATA command) that contains the following MAXIMUM\_UNASSIGNED\_WORKUNITS statement:

```
MAXIMUM_UNASSIGNED_WORKUNITS= 0
```

Once cleanup processing is complete, the DASD management subsystem may be enabled by removing the override configuration file and issuing the DIRMaint RLDDData command.

5. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.

## CLONEDISK



### Purpose

Use the CLONEDISK operand of the DIRMAINT command to do either of the following:

- Create a new minidisk by cloning another existing minidisk. The new disk will have the same device type and size as the existing disk, and will also contain all of the data from the existing disk. (The existing disk is not changed.)
- Replace all of the data on one existing disk with the data from another existing disk of the same device type and size.

### Operands

#### **vaddr**

is the virtual device address of the new minidisk.

#### **source\_owner**

is the name of the user, identity, or subconfiguration directory entry in which the minidisk being cloned exists,

#### **source\_addr**

is the specific minidisk (owned by *source\_owner*) being cloned.

#### **startloc**

is a decimal cylinder relocation factor or a decimal FB-512 device block number that specifies the beginning of the minidisk extent corresponding to cylinder 0 or block 0 of the user's minidisk.

It is not advisable to define a minidisk starting at real cylinder 0, unless it is to be used by operating system (OS) indexed sequential access method (ISAM), in which case it must begin at real cylinder 0. If you do assign a minidisk beginning at real cylinder 0, the user who owns it must realize that the minidisk label is the real label that the user and the system use to identify the disk. Remember that CP-owned volumes must not have minidisks beginning at real cylinder 0. Generally, CKD DASD extents should begin with cylinder 1 or higher, and FBA DASD extents should begin with block 32 or higher.

**AUTOG**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a volume within a specified group. The group must have been defined in the EXTENT CONTROL file, and the regions within the group must also be defined.

**AUTOR**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a specified region or volume. The region must have been defined in the EXTENT CONTROL file.

**AUTOV**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a specified volume. The volume must have been defined in the EXTENT CONTROL file.

***valid***

specifies the volume label of the DASD volume, 1 to 6 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +, on which the minidisk is to be allocated, or the reserved identifier of &SYSRES. This applies to fixed allocation (where a specific starting cylinder or block number is specified) and to automatic allocation on a volume (AUTOV).

***groupname***

specifies the name of the DASD allocation group from which DirMaint will pick the *volser* and starting CKD cylinder or FBA block. Groups consist of one or more specific *regions*. This is valid for AUTOG requests only. Groups must be defined in the EXTENT CONTROL file, and can be 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +.

***regionname***

specifies the name associated with the *region*, 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +, on which the minidisk is to be allocated. This is valid for AUTOR requests. The named region must be defined in the REGIONS section of the EXTENT CONTROL file. The region defines a segment on a specific DASD volume for allocation.

**DEVNO *raddr***

specifies a full volume minidisk.

***raddr***

specifies the real device number of the full volume minidisk.

***mode***

specifies the user's mode of access at logon. If the minidisk owner allows other users access to the minidisk by establishing access passwords, then the access conditions may be modified by such links. Valid access modes are:

**Mode****Definition****R**

Read-only (R/O) access is desired. The access is not given if any other user has a link to the disk in write or exclusive status.

**RR**

Read-only (R/O) access is desired, even if another user has a link to the disk in write status. Access is denied if any user has an exclusive link.

**W**

Write access is desired. The disk is not accessible if any other user has a link to the disk in read or write status.

**WR**

Write access is desired, but read-only access is acceptable if any other user has a link to the disk in read or write status. Access is denied if any user has an exclusive link.

**M**

Multiple access is desired. A write link to the disk is given unless another user already has write or any exclusive access to it; in which case, the disk is not accessible.

**MR**

Write access to the minidisk is desired unless another user already has write or any exclusive access to it. In this case, read-only access to the minidisk is allowed. Access is denied if any user has an exclusive link.

**MW**

Unconditional write access to the disk is desired. Access is denied if any user has an exclusive or stable link.

***suffix***

specifies an optional mode suffix to be directly appended to the access mode. The suffix letters can be combined as follows (note that while CP requires they be specified in this order, DirMaint allows them in any order and will reorder them appropriately):

1. V or null, plus
2. S, E, or null (S and E are mutually exclusive), plus
3. D or null.

Therefore, the only valid combinations are: V, S, E, D, VS, VE, VD, VSD, VED, SD, or ED. This suffix is in turn concatenated with the mode, with no intervening blanks. For example, RV, RRS, WVE, and MVSD are all valid.

**Suffix****Definition****V**

Tells CP to use its virtual reserve/release support in the I/O operations for the minidisk.

**S**

Authorizes the virtual machine to use the LINK command stable access modes (SR, SW, SM) against the specified user's minidisk.

**E**

Authorizes the virtual machine to use the LINK command exclusive and stable access modes (ER, EW, SR, SW, SM) against the specified user's minidisk.

**D**

Tells CP that the device should not be defined when the virtual machine initially logs on or is autologged, but to defer doing so until an explicit LINK command is issued for that device.

**PWs**

informs DirMaint that the following parameters are to be considered passwords.

***readpass***

is a password that allows a user other than the owner to get a read-only (R/O) link to the minidisk if the other user knows this 1 to 8 character password.

***writepass***

is a password that allows a user other than the owner to get a WRITE link to the minidisk if the other user knows this 1 to 8 character password.

***multipass***

is a password that allows a user other than the owner to get a MULTIPLE link to the minidisk if the other user knows this 1 to 8 character password.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. If the specified *vaddr* already exists, then the MDISK Allocation Parameters must be omitted, and the existing disk must be the same device type and size as the source disk being cloned.
3. If the specified *vaddr* does not already exist, then the MDISK Allocation Parameters must be specified. The source disk device type and size will be used in the allocation of the new disk.
4. Access mode and access passwords can be set and changed by the owner. If they are not entered with the CLONEDISK operand, the CP default will be access mode W, with no access passwords.
5. Due to the syntax of the CP MDISK statement, the use of access passwords requires that a mode be specified.
6. You may deliberately want to allocate a specific extent on a volume. If this extent would cross region boundaries or occupy a reserved area, you must have issued the EXTNCHK OFF operand before entering the CLONEDISK operand.
7. A profile directory entry may not be the target of the CLONEDISK operand.
8. When the *valid* used on the CLONEDISK command does not exist in the EXTENT CONTROL file the volume will be allocated explicitly. The devtype value provided (3390, 3380) will be used as the default device type. If the volume is attached to the system, DirMaint will query the device to determine the maximum size. In an SSI environment, the queries are issued on each SSI member and the maximum size of the query responses is used. When the volume is not attached, DirMaint will use the default maximum size that is found in the DEFAULTS section of the EXTENT CONTROL file or the DEFAULTS DATADVH file (for example, a 3390 device will default to a maximum allocatable cylinder value of 1113).
9. If a DASD management work unit associated with this command becomes stalled or hung on the DATAMOVE machine, manual intervention may be necessary in order for the work unit to complete. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.
10. The IBM-supplied default FLASHCOPY support in the DATAMOVE\_DDR\_EXIT (DVHDXD) can be modified using the DVHDXD\_FLASHCOPY\_BEHAVIOR=, DVHDXD\_FLASHCOPY\_COMPLETION\_WAIT=, and DVHDXD\_FLASHCOPY\_NOREPLY\_WAIT= configuration statements. For more information on these statements, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*.
11. CLONEDISK to a space-efficient target disk is not available, as DirMaint does not implement IBM FlashCopy/SE support and a DDR copy to all extents of the space-efficient target would defeat the purpose of space-efficient DASD (which is to copy as little data as possible).
12. If the AT prefix is specified for a subconfiguration entry when the CLONEDISK operand is issued, the AT keyword value will be used to select the DATAMOVE machine configured at the specified node. In this respect, the FORuser prefix keyword must specify the subconfiguration entry name in order to modify a subconfiguration entry with the CLONEDISK operand.
13. If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## Examples

1. You've just added a new CMS user like an existing CMS user. The template for the new user did not contain an MDISK 191 statement. You now wish to create a 191 MDISK for this new user, using space from the CMSUSERS group, and copy some starter files to that disk, such as a PROFILE EXEC and a PROFILE XEDIT. These sample files are on CMSSAMPL's 191 disk. You want the default access mode to be read/write, even if someone else has it linked for read access, with read/only access if someone else has it linked for write access. You do not want any link passwords. Issue the following command:

```
DIRM FOR NEWUSER CLONEDISK 191 CMSSAMPL 191 AUTOG CMSUSERS MR
```

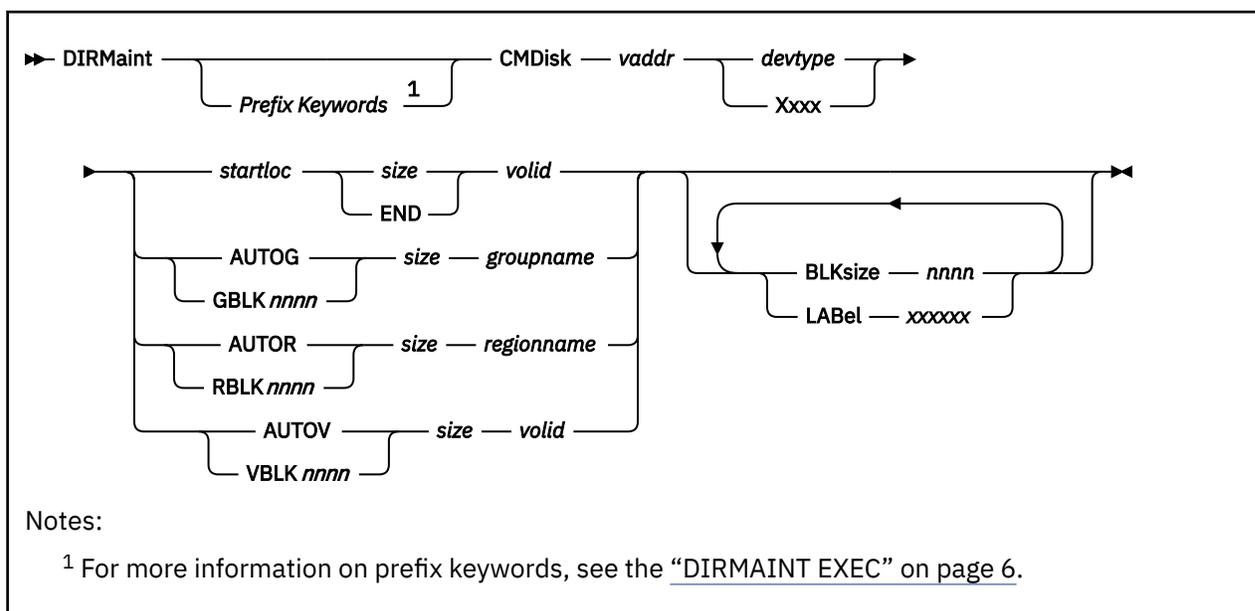
2. You've just finished teaching a course on Linux. Each student had their own Linux virtual machine. These machines now need to be reset to the clean install state in preparation for the next course. (Existing disk space is being re-used.) Issue commands like the following:

```
DIRM FOR LINUX001 CLONEDISK 191 LINMASTR 191
DIRM FOR LINUX001 CLONEDISK 150 LINMASTR 150
DIRM FOR LINUX001 CLONEDISK 151 LINMASTR 151
```

3. You're providing internet services and Linux servers for your customers. You have just created a new Linux server, and now need to allocate disk space and “install” Linux for the customer. The space comes from the LINUSERS pool. Again, you want MR access with no passwords. Issue commands like the following:

```
DIRM FOR LINUX111 CLONEDISK 191 LINMASTER 191 AUTOG LINUSERS MR
DIRM FOR LINUX111 CLONEDISK 150 LINMASTER 150 AUTOG LINUSERS MR
DIRM FOR LINUX111 CLONEDISK 151 LINMASTER 151 AUTOG LINUSERS MR
```

## CMDISK



The CMDISK operand of the DIRMAINT command changes an existing minidisk in terms of residency, size, and extent occupied.

The disk must be formatted as a CMS disk and must not be RECOMPed or RESERVED. DirMaint will format the new extent and move existing files to the new extent. If no files are involved, refer to DIRM RMDISK to alter an existing extent.

See Usage Note [“9”](#) on page 61 for DirMaint-owned minidisks.

### Operands

#### **vaddr**

is the virtual device address of the minidisk

#### **devtype**

is the device type of the volume to which the minidisk is assigned. Otherwise, if automatic allocation is used, specify the device type as *Xxxx*. The automatic allocation options include these operations:

- AUTOG and GBLK
- AUTOR and RBLK
- AUTOV and VBLK

The real device type will be taken from the REGIONS section in the EXTENT CONTROL file for that volser.

#### **startloc**

is a decimal cylinder relocation factor or a decimal FB-512 device block number that specifies the beginning of the minidisk extent corresponding to cylinder 0 or block 0 of the user's minidisk.

#### **AUTOG**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a volume within a specified group. The group must have been defined in the EXTENT CONTROL file, and the regions within the group must also be defined.

#### **GBLKnnnn**

is functionally equivalent to AUTOG except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: GBLK512, GBLK0512, GBLK1024, GBLK2048, GBLK4096, GBLK1K, GBLK2K, and GBLK4K. In addition, GBLK800 or GBLK0800 may be specified for device types 3350, 3375, and 3380.

If the existing disk is permanently allocated and has already been formatted for CMS use, the new extent will actually be formatted with the same block size as the existing disk. However, DirMaint will use the block size specified to compute the number of CKD cylinders necessary to satisfy the request.

**AUTOR**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a specified region or volume. The region must have been defined in the EXTENT CONTROL file.

**RBLKnnnn**

is functionally equivalent to AUTOR except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: RBLK512, RBLK0512, RBLK1024, RBLK2048, RBLK4096, RBLK1K, RBLK2K, and RBLK4K. In addition, RBLK800 or RBLK0800 may be specified for device types 3350, 3375, and 3380.

**AUTOV**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a specified volume. The volume must have been defined in the EXTENT CONTROL file.

**VBLKnnnn**

is functionally equivalent to AUTOV except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: VBLK512, VBLK0512, VBLK1024, VBLK2048, VBLK4096, VBLK1K, VBLK2K, and VBLK4K. In addition, VBLK800 or VBLK0800 may be specified for device types 3350, 3375, and 3380.

If the existing disk is permanently allocated and has already been formatted for CMS use, the new extent will actually be formatted with the same block size as the existing disk. However, DirMaint will use the block size specified to compute the number of CKD cylinders necessary to satisfy the request.

**size**

is a 1 to 8 digit decimal number specifying the size of the new minidisk in CKD cylinders or FBA blocks. For allocation requests involving block sizes, allocation will be converted to the exact number of FB-512 blocks for FBA devices or to the minimum number of cylinders needed to contain the specified number of CMS blocks for CKD devices.

**END**

specifies that the minidisk will extend from the assigned starting cylinder or block to the end (last cylinder or block) of the volume in which it is defined. This parameter is valid for fixed allocation only.

**valid**

specifies the volume label of the DASD volume, 1 to 6 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +, on which the minidisk is to be allocated, or the reserved identifier of &SYSRES. This applies to fixed allocation (where a specific starting cylinder or block number is specified) and to automatic allocation on a volume (AUTOV, VBLKnnnn).

**groupname**

specifies the name of the DASD allocation group from which DirMaint will pick the *volser* and starting CKD cylinder or FBA block. Groups consist of one or more specific *regions*. This is valid for AUTOV and GBLKnnnn requests only. Groups must be defined in the EXTENT CONTROL file, and can be 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +.

**regionname**

specifies the name associated with the *region*, 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +, on which the minidisk is to be allocated. This is valid for AUTOR and RBLKnnnn requests. The named region must be defined in the REGIONS section of the EXTENT CONTROL file. The region defines a segment on a specific DASD volume for allocation.

**BLKsize nnnn**

specifies the block size to use when formatting the new extent. The valid values include:

512

1024

2048  
4096  
1K  
2K  
4K

If the BLKSIZE parameter is not specified, but the LABEL parameter is specified, the disk will be formatted with the default blocksize determined by the CMS FORMAT command. If neither BLKSIZE nor LABEL are specified, the disk will not be formatted.

If the BLKSIZE parameter is not specified, the new extent will be formatted with the same blocksize as the old extent.

#### **LAbel xxxxxx**

specifies the minidisk label to use when formatting the new extent. The labels are 1 to 6 alphanumeric characters.

If the LABEL parameter is not specified, the new extent will be given the same label as the old extent.

### **Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRextxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. It is not advisable to define a minidisk starting at real cylinder 0 unless it is to be used by OS ISAM. If it is to be used by OS ISAM, it must begin at real cylinder 0. If you do assign a minidisk beginning at real cylinder 0, the user who owns it must be aware that the minidisk label is the real label that the user and the system use to identify the disk. CP-owned volumes must not have minidisks beginning at real cylinder 0. Generally, CKD DASD extents should begin with cylinder 1 or higher, and FBA DASD extents should begin with block 32 or higher.
3. In using this operand be aware that another user may be linked to the minidisk being changed. It is your responsibility to have that link removed and re-established after the change if necessary.
4. This operand is not used to change the access mode or access passwords. The owner of the minidisk can use the MDISK operand to do that directly.
5. The CMDISK operand can be used to change the size (number of cylinders or blocks) of a minidisk. This requires that a new extent be allocated, the files on the old extent copied to the new, and finally that the old extent be released. Data movement is not always automatic. See Usage Note [“8” on page 61](#).
6. The CMDISK operand can also be used to move the extent to a new residence without changing the size.
7. Converting a T-DISK minidisk to a permanent allocation requires the use of DIRM RMDISK.
8. When data must be moved from the old extent to the new extent, the move will be coordinated by a DIRMAINT virtual machine. The actual movement of data is done automatically by a DATAMOVE virtual machine.
  - a. The new extent is allocated, but attached to a user defined as a DATAMOVE machine in the CONFIG\* DATADVH file. The role of this user is to do the copying and reformatting operations.

**Note:** You may define several DATAMOVE machines.

- b. The user's original minidisk is also transferred to a user defined as a DATAMOVE machine in the CONFIG\* DATADVH file. The role of this user is to do the copying and reformatting operations.

For more information on defining a DATAMOVE machine in the CONFIG\* DATADVH file, see [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#).

9. Do not attempt to move DirMaint minidisks or RACF<sup>®</sup> owned minidisks with the CMDISK operand. This must be done manually using this procedure:

For each minidisk to be moved, use the AMDISK function to allocate a new extent as desired for the minidisk to be moved. Use a virtual address that you can relate to the basic MDISK. For example, if you are moving the DIRMAINT 1DB minidisk, allocate a new 5DB.

After the newly-allocated extents are really available:

- a. Use the OFFLINE operand. This prevents updating the online directory during the next few steps.
- b. Use the GET operand. Edit the returned file (for example DIRMAINT DIRECT) and swap the virtual addresses of the minidisks.

Original Extent:

```
MDISK 1DB 3390 123 005 vo1001 MR
```

New Extent:

```
MDISK 5DB 3390 487 006 vo1016 MR
```

Change as follows:

Original Extent:

```
MDISK 5DB 3390 123 005 vo1001 MR
```

New Extent:

```
MDISK 1DB 3390 487 006 vo1016 MR
```

- c. Use the REPLACE operand. This updates the USER DIRECT file to map DirMaint's minidisks to the new extents. The OFFLINE operand prevents it from being placed online.
- d. Use the SHUTDOWN operand. DirMaint logs off, but the online directory still maps DirMaint's minidisks to the original extent(s). The source directory, however, reflects the newly-allocated extents as current.
- e. Log on to the user ID that owns the minidisk you are increasing. In this example, log on to the DIRMAINT service machine.
- f. Format the newly-allocated minidisk(s), for example, 5DB.
- g. Access the original minidisk (for example, 1DB) as a free filemode (for example, as filemode A). Then copy all files from the original minidisk (1DB – filemode A) to the newly-allocated and formatted minidisk; for example, to 5DB. Do this for all minidisks being moved at this time. When all minidisks have been copied to the newly-allocated extents, continue with the next step.
- h. If you are not logged on to the DIRMAINT server, log off the user ID that owns the newly-updated minidisks. Log on to the DIRMAINT user ID, if necessary.
- i. Manually invoke the system command to update the object directory using DIRECTXA. The source directory already contains the new minidisks. The new online directory contains mapping to the newly-allocated minidisks. These new extents are mirror images of the original extents, as shown in the example above. If you are running in an SSI cluster, you should log on to DIRMAINT on each system in your SSI cluster and use DIRECTXA to update the object directory online on each member.
- j. Log off, then as an authorized user, AUTOLOG the DIRMAINT service machine. The DIRMAINT machine will be operational using the new extents. When you are sure the DIRMAINT machine is operating correctly, use the ONLINE operand to re-enable execution of DIRECT. Then use the DMDISK operand to delete the original extents (now 5DB, and so on).
- k. As an authorized user, FORCE and AUTOLOG the DATAMOVE user ID. If you are running a multi-member SSI cluster, FORCE and AUTOLOG all DATAMOV\* and DIRMSAT\* server machines on all members.

The purpose of this procedure is to preserve the integrity of DirMaint as an operational entity, and to ensure that any unrelated directory changes made during the time span required to perform the first few steps are preserved.

10. The CMDISK operand cannot be used to change an existing DEVNO, T-DISK, or V-DISK minidisk. The CMDISK operand will be rejected if the existing minidisk being changed is a DEVNO, T-DISK, or V-DISK minidisk. The RMDISK operand may be used to change the minidisk definition, but does not cause any data movement to take place. Alternatively, DMDISK, followed by AMDISK, or GET and REPLACE, may be used to modify the definition of a DEVNO, T-DISK, or V-DISK minidisk.
11. If the minidisk is not a CMS formatted disk or if it is either RESERVED or RECOMPed, use AMDISK to add a new disk of the required type and size, have the owning user copy the existing data to the new disk then use DMDISK with the KEEPLINKS option to destroy the old extent. Follow this by a CHVADDR to reassign the new disk to the old disks address.
12. If a DASD management work unit associated with this command becomes stalled or hung on the DATAMOVE machine, manual intervention may be necessary in order for the work unit to complete. See *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.
13. If the AT prefix is specified for a subconfiguration entry when the CMDISK operand is issued, the AT keyword value will be used to select the DATAMOVE machine configured at the specified node. In this respect, the FORuser prefix keyword must specify the subconfiguration entry name in order to modify a subconfiguration entry with the CMDISK operand.
14. If your system specifically has RACF installed as the ESM, see *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## Cancelling a DIRMAINT CMDISK Request

Many of the situations involving manual intervention in previous releases of DirMaint have been handled automatically in this release. Situations that result in DATAMOVE's inability to copy data (Non-CMS formatted minidisk or nonformatted minidisk for example) cause DIRMAINT to automatically rollback the CMDISK request and return the extent to the original owner.

There are situations where a user-initiated cancel of a CMDISK request may be required. If active links exist to the source of a CMDISK, DATAMOVE is forced to wait until these links are detached. When the links are detached the DATAMOVE machine will continue with its operation. Should the user wish to cancel the transaction while this wait is in progress the following steps should be used.

1. Find the work unit number associated with your transaction.
 

This can be done using the DIRM STATUS and DIRM QUERY commands. Each workunit contains information on who issued the command and what the command was.
2. Use the DIRM WORKUNIT command to CANCEL the appropriate workunit.
 

This will cause DATAMOVE to give up waiting on the active links and return control of the work unit to DIRMAINT. DIRMAINT will rollback (return the extent to the user and release all locks) and deallocate the workunit.

Should a work unit fail for some reason, DirMaint is designed to automatically rollback the transaction for the most common failures. Failures considered too dangerous for automatic rollback are left for administrative review and action. For more information on automatic and manual rollback of workunits, see *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*.

## Examples

1. DIRM FOR JONES CMD 193 FB-512 1672000 880 VOL005

Changes the 193 minidisk belonging to JONES, from whatever and wherever it is, to 880 blocks. It is to be allocated on a 9336 or other FBA volume, valid=VOL005, at real block 1672000. The current access mode and access passwords, if any, remain unchanged.

2. DIRM FOR BILL CMD 192 X AUTOV 10 VOLXYZ

Specifies that the 192 minidisk belonging to user ID BILL be changed to 10 cylinders. The allocation is to be made on volume valid=VOLXYZ.

**3. DIRM FOR STEVE CMD 186 3390 VBLK512 1000 V3390A**

This command tells DirMaint to change STEVE's existing 186 minidisk, whatever its existing device type and location to 2 cylinders (1470 blocks total) on 3390 volume V3390A. The new disk will have the same CMS block format as the old disk: 990 blocks if formatted with BLKSIZE 1024, 630 blocks if formatted with BLKSIZE 2048, or 360 blocks for BLKSIZE 4096. For a non-CMS disk, the administrator must copy the data and use the appropriate DIRMAINT commands to let DirMaint know when to make the new extent available to the user. But the size is still computed the same: 1000 blocks requested, divided by 735 512-byte blocks per 3390 cylinder, equals 1.36 cylinders, which is increased to 2 cylinders to satisfy the minimum requirement.

**4. DIRM FOR HARRY CMD 23F X AUTOG 8 MSSVOLS**

Requests that HARRY's 23F minidisk be changed to 8 cylinders. This is like the AUTOV example, except that the allocation is to be made automatically on a region within group MSSVOLS. The group name, MSSVOLS, and all regions within it, must be currently defined in the EXTENT CONTROL file.

## CMS

► DIRMaint — *Prefix Keywords* <sup>1</sup> — CMS — *command* — ◄

Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the CMS operand of the DIRMAINT command to execute a CMS command string on the DIRMAINT server.

### Operands

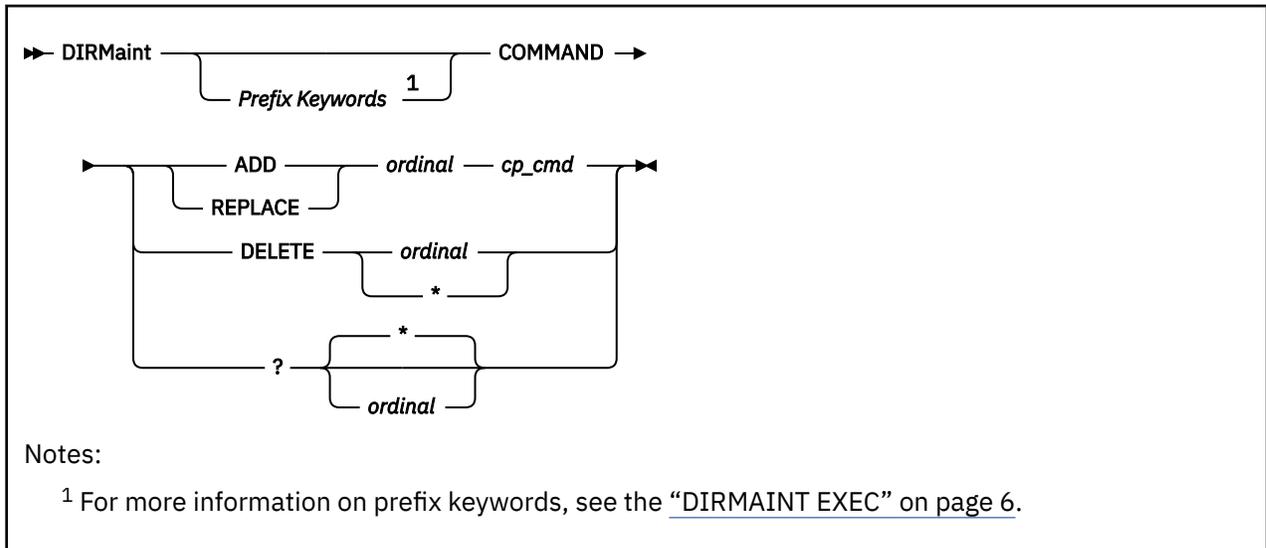
#### *command*

specifies the command to be executed on the DIRMAINT server.

### Usage Notes

You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.

## COMMAND



### Purpose

Use the COMMAND operand of the DIRMaint command to add, replace, delete, or query the COMMAND statements in a directory entry. COMMAND statements are added, replaced, and deleted using an ordinal value among COMMAND statements in the directory entry.

### Operands

#### *cp\_cmd*

is the CP command to be added or replaced.

**Note:** The total length of the CP command must be less than or equal to 220 characters. Also, due to line continuation rules, the length of any single token in the CP command must be less than or equal to 70 characters.

#### *ordinal*

is the ordinal value within the directory entry COMMAND statements.

#### **ADD**

adds the specified CP command at the given ordinal value within the directory entry COMMAND statements.

#### **REPLACE**

replaces the COMMAND statement at the given ordinal value within the directory entry COMMAND statements with the specified CP command.

#### **DELETE**

deletes all COMMAND statements in the directory entry by specifying '\*' or deletes the COMMAND statement at the specified ordinal value within the directory entry COMMAND statements.

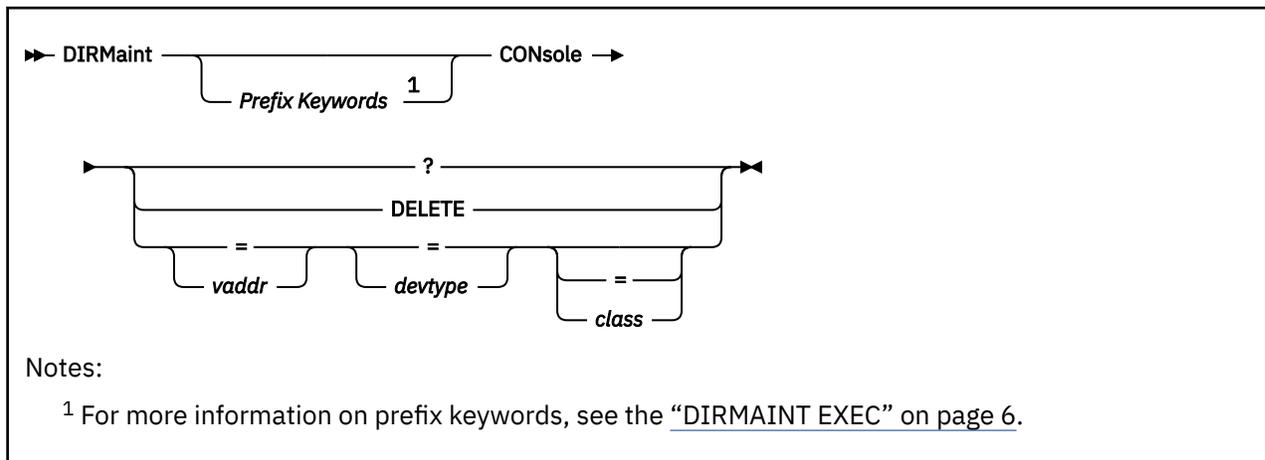
#### **?**

queries all COMMAND statements in the directory entry by specifying '\*' (this is the default) or queries the COMMAND statement at the specified ordinal value within the directory entry COMMAND statements.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.

## CONSOLE



### Purpose

Use the CONSOLE operand of the DIRMAINT command to create, delete, or query the CONSOLE statement in a directory entry.

### Operands

?

specifies that the CONSOLE statement is to be queried.

#### DELETE

specifies that the CONSOLE statement is to be deleted.

=

specifies the value currently contained in the position is to remain.

#### vaddr

specifies the virtual address of the console.

#### devtype

specifies the device type. Valid device types are 1052, 3210, 3215, and 3270.

#### class

specifies the 1 character spooling class.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For a complete description of the CONSOLE directory statement, see [z/VM: CP Planning and Administration](#).
3. To alter the ID of the secondary console, see [“SECUSER”](#) on page 232.
4. The equal (=) parameter implies that the token in that position already exists. If the effective CONSOLE statement does not have a parameter in the position denoted by the equal sign, the request will be rejected.
5. In level 140A the CONSOLE command performs the GETCONSOLE function. In level 150A the CONSOLE command changes the device characteristics of the virtual machine's CONSOLE directory statement. For more information, see [“GETCONSOLE”](#) on page 121.

## CONSOLE

6. If an external security manager (ESM) is installed and security label checking is enabled, the secondary user or observer defined may not be functional. For additional information, contact your security administrator.

## CP

►► DIRMaint — *Prefix Keywords* <sup>1</sup> — CP — *command* ►►

Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the CP operand of the DIRMAINT command to execute a CP command string on the DIRMAINT server.

### Operands

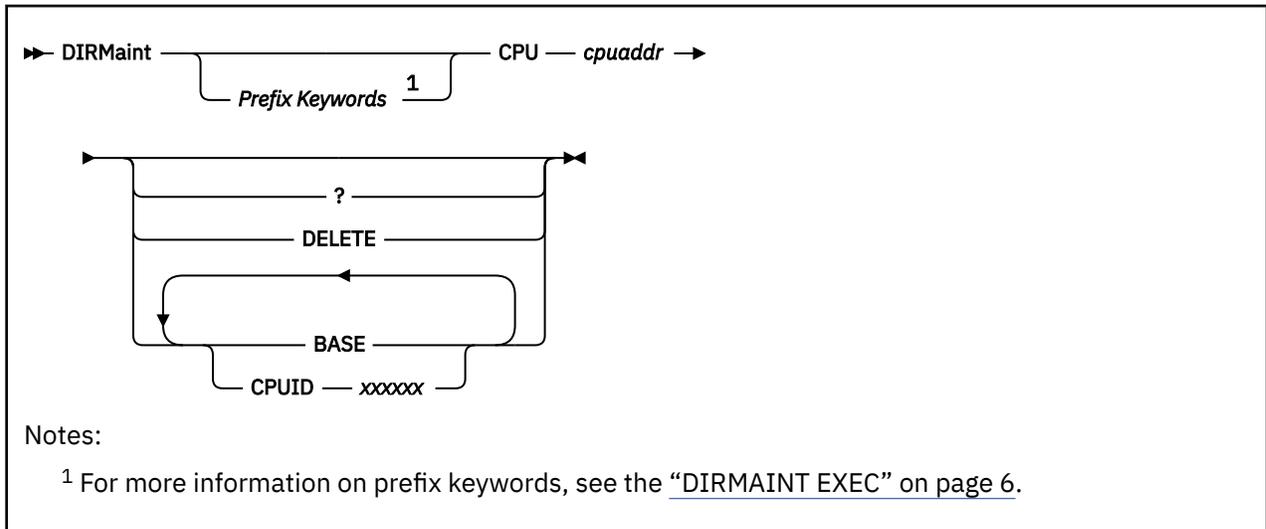
#### *command*

specifies the command to be executed on the DIRMAINT server.

### Usage Notes

You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.

## CPU



### Purpose

Use the CPU operand of the DIRMAINT command to allow the general user to add, change, delete, or query the central processing unit (CPU) statement in a profile, user, or identity entry.

### Operands

#### *cpuaddr*

specifies the address of the virtual processor being defined, changed, or deleted. The processor address can be any hexadecimal value from 0 to 3F.

#### ?

issues a query against the specified CPU statement.

#### DELETE

deletes the specified CPU statement from the directory entry.

#### BASE

specifies that the CPU identified on this statement is defined as the base CPU.

#### CPUID *xxxxxx*

specifies the processor ID number that is stored in bits 8 to 31 of the CPU ID and is returned in response to the STIDP instruction.

### Usage Notes

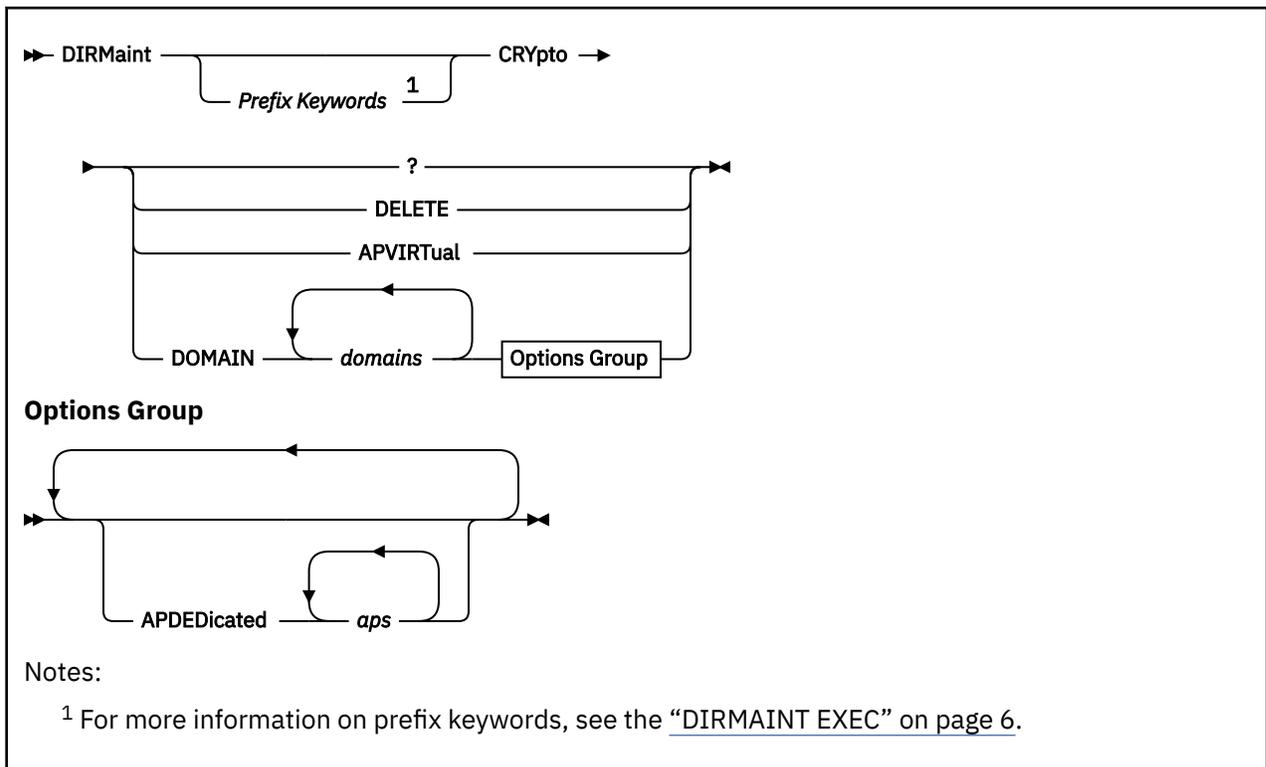
1. You can use one or more of the prefix keywords (REQest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the CPU directory statement, see [z/VM: CP Planning and Administration](#).
3. The CPU operand is equivalent to the SETCPU operand.
4. Successful execution of the DIRMAINT CPU operand depends on the MACHINE directory statement setting defining the maximum number of virtual processors allowed to be defined (indicated by the *mcpu* value). These rules apply:
  - Modification of an existing CPU statement is always valid. However, only one CPU statement can be given the BASE attribute.
  - Deletion of an existing CPU statement is always valid.

- If a MACHINE statement does exist and the *mcpu* field contains a value, this value is the maximum number of allowable CPU statements.
- If a MACHINE statement does not exist, or does not specify an MCPU value, only one CPU may be defined.

If any of these rules are violated, CPU operand processing will terminate and an appropriate error message is issued.

5. A new CPU statement is always added after all existing CPU statements in your directory entry unless your installation has chosen to sort by device addresses. The order of CPU statements determines the assignment of real processors to virtual CPUs.
6. The CRYPTO operand is still accepted for compatibility purposes, but provides no function.

## CRYPTO



## Purpose

Use the CRYPTO operand of the DIRMAINT command to add, delete, or query the current CRYPTO statement specified in a user, identity, or subconfiguration entry.

## Operands

?

specifies that the existing CRYPTO statement is to be queried.

### DELETE

specifies that the existing CRYPTO statement is to be deleted from the directory entry.

### APVIRTual

tells CP that this virtual machine may have access to the system's shared crypto resources. The APVIRTUAL operand cannot be specified if the APDEDICATED operand is specified.

### DOMAIN *domains*

specifies up to 256 domains the virtual machine may use. Domains are 0 through 255, specified in decimal. They can appear in any order, without duplication. The DOMAIN operand can be specified more than once on additional CRYPTO statements, but the DOMAIN operand must be specified before APDEDICATED is specified.

The DOMAIN operand should be specified only with APDEDICATED. The DOMAIN operand should not be specified alone or with APVIRTUAL. Although this is accepted for compatibility reasons, it provides no function.

### APDEDicated *aps*

specifies up to 256 crypto adapters the virtual machine may use for dedicated access. Adapter numbers are 0 through 255, specified in decimal. They may appear in any order, without duplication. The DOMAIN operand also must be specified to indicate the crypto domains to access. The crypto domains equivalent to the domains specified by all of the DOMAIN operands will be assigned

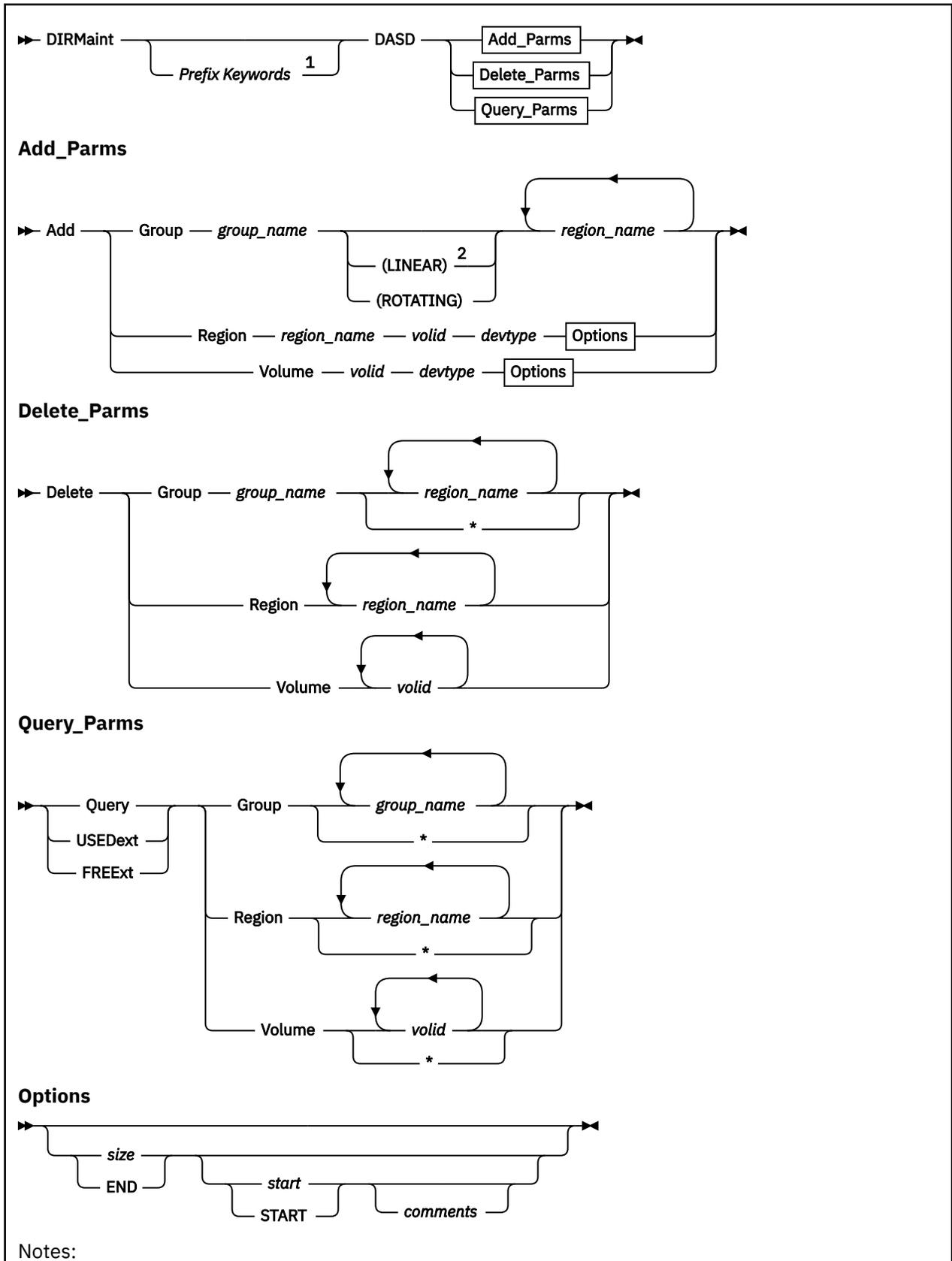
to the virtual machine for each adapter specified on the APDEDICATED operand on all CRYPTO statements. The APDEDICATED operand cannot be specified if the APVIRTUAL operand is specified. The APDEDICATED operand can be specified more than once on additional CRYPTO statements.

The crypto adapters specified must be selected from the set of cryptos selected on the Cryptographic Online List on the Crypto Image Profile Page for the logical partition in which z/VM is running. The DOMAINS specified must be selected from the set of domains specified on the Usage Domain Index selections on the Crypto Image Profile Page for the Logical Partition. For more information about the Crypto Image Profile Page, see the *Processor Resource/Systems Manager Planning Guide*, SB10-7041.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the CRYPTO directory statement, see [z/VM: CP Planning and Administration](#).
3. The DOMAIN operand should not be specified alone or with APVIRTUAL, although it is accepted for compatibility but provides no function. The DOMAIN operand should be specified with APDEDICATED.
4. The CSU, KEYENTRY, SPECIAL, and MODIFY operands are still accepted for compatibility purposes, but provide no function.

# DASD



<sup>1</sup> For more information on prefix keywords, see the “DIRMAINT EXEC” on page 6.

<sup>2</sup> Neither (LINEAR) nor (ROTATING) is permitted for an existing group. (LINEAR) is the default for a new group.

## Purpose

Use the DASD operand of the DIRMAINT command to synchronously update the DASD management EXTENT CONTROL file.

## Operands

### *group\_name*

Specifies the name of this group entry. Specification of this field is subject to the following rules:

- If a group entry shares the same name with another group entry, it is considered a single group. For instance, consider the following groups segment from an EXTENT CONTROL file:

```
MyGroup  Mike1 Mike2
MyGroup2 Mark1
MyGroup  Mike3
```

This series of statements actually defines two groups, MyGroup and MyGroup2. The same information could have been represented as:

```
MyGroup  Mike1 Mike2 Mike3
MyGroup2 Mark1
```

This gives the user the ability to define large groups without using excessively long records.

- Group names must not start with a valid EXTENT CONTROL file tag.
- Group names may consist of characters from the following list: A-Z, 0-9, #, @, \$, :, - and +.
- DirMaint requires that this field be eight characters or less.

### (LINEAR)

Specifies the linear scanning method, in which the first region within a group is scanned for allocation until full, then the second region, and so on, up until the last region.

### (ROTATING)

Specifies the rotating scanning method, in which the first region within a group is scanned for the first allocation, then the second region for the second allocation, and so on, with each new allocation starting at the next region.

### *region\_name*

Specifies the name of this region entry. Specification of this field is subject to the following rules:

- Region names must be unique. If a region entry shares the same name with another region entry, the first record is used, the second entry is ignored.
- Region names may consist of characters from the following list: A-Z, 0-9, #, @, \$, :, - and +. DirMaint requires that this field be eight characters or less.

### *valid*

Specifies the volume ID of the region where the region is located.

- This value represents the value placed into the *valid* field on any minidisks generated from this region.
- Volume IDs supported by DirMaint consist of characters from the following list: A-Z, 0-9, #, @, \$, :, - and +.

### *devtype*

Specifies the DASD type associated with each entry.

**size**

Specifies the block size of this entry. A keyword of END can be used to specify the end (last cylinder) of the volume.

Note that if the VOLUME is not defined to the system, you need to supply both the *size* and *start* values. When adding a VOLUME, the *size* value must be greater than the *start* value.

The *regend* value is calculated and updated in the EXTENT CONTROL file as:

```
regend = (start + size) - 1
```

The *volend* value is calculated and updated in the EXTENT CONTROL file as:

```
volend = size - 1
```

**start**

Specifies the starting block or cylinder (inclusive) of the region. A keyword of START can be used to define the start of a region. This translates to cylinder 1 of a CKD device and block 32 of an FBA device.

Note that if the VOLUME is not defined to the system, you need to supply both the *size* and *start* values. When adding a VOLUME, the *size* value must be greater than the *start* value.

**comments**

Additional comments that will be included in the reports. (See [“Reports”](#) on page 76.)

**USEDext**

Specifies that a listing should be generated showing allocated extents on volumes, groups, and regions used to contain minidisks. For more information on USEDext, see [“USEDEXT”](#) on page 281.

**FREEExt**

Specifies that a listing should be generated showing free extents on volumes, groups, and regions used to contain minidisks. For more information on FREEExt, see [“FREEEXT”](#) on page 117.

**Reports**

The reports returned by the QUERY, USEDext, and FREEExt options will appear in the following formats:

**QUERY GROUP**

```
DVHSD3561I GROUP=group_name ALLOCATE={LINEAR|ROTATING} REGIONS=nn
DVHSD3562I GROUP=group_name REGIONS= reg1 reg2 reg3 ... regnn
```

Unless the RETVIA/RETURNVIA RDRFILE prefix keyword is used, the response will be returned in the message format shown above. There will be one 3561 message per group, and as many 3562 messages as needed to list all of the regions in the group.

If the RETVIA/RETURNVIA RDRFILE prefix keyword is used, the result will be a reader file containing the above format *without* the message number ('DVHSD3561I' and 'DVHSD3562I') prefixes. The name of the file returned is dependent on the options specified on the command, as follows:

**group\_name QUERY**

will be returned if one specific group name was specified on the command.

**ALLGRPS QUERY**

will be returned if the command was issued with a group name of '\*'.

**MULTIGRP QUERY**

will be returned if the command was issued with more than one group name.

**QUERY REGION**

```
DVHSD3563I REGION VOLUME START SIZE END DEV-TYPE COMMENTS
DVHSD3564I xxxxxxxx validx nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn dddddddd etc.
DVHSD3564I xxxxxxxx validx nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn dddddddd etc.
```

Unless the RETVIA/RETURNVIA RDRFILE prefix keyword is used, the response will be returned in the message format shown above. There will be one 3563 message per query request, and one 3564 message per each region. The SIZE column is calculated as (END-START)+1.

If the RETVIA/RETURNVIA RDRFILE prefix keyword is used, the result will be a reader file containing the above format *without* the message number ('DVHSD3563I' and 'DVHSD3564I') prefixes. The name of the file returned is dependent on the options specified on the command, as follows:

#### **region\_name QUERY**

will be returned if one specific region name was specified on the command.

#### **ALLREGS QUERY**

will be returned if the command was issued with a region name of '\*'.

#### **MULTIREG QUERY**

will be returned if the command was issued with more than one region name.

#### **QUERY VOLUME**

```
DVHSD3565I VOLUME DEV-TYPE  SIZE      END
DVHSD3566I xxxxxxxx dddddddd nnnnnnnnnn nnnnnnnnnn
DVHSD3566I xxxxxxxx dddddddd nnnnnnnnnn nnnnnnnnnn
```

Unless the RETVIA/RETURNVIA RDRFILE prefix keyword is used, the response will be returned in the message format shown above. There will be one 3565 message per query request, and one 3566 message per each volume. The END column is calculated as SIZE-1.

If the RETVIA/RETURNVIA RDRFILE prefix keyword is used, the result will be a reader file containing the above format *without* the message number ('DVHSD3565I' and 'DVHSD3566I') prefixes. The name of the file returned is dependent on the options specified on the command, as follows:

#### **volume\_name QUERY**

will be returned if one specific volume name was specified on the command.

#### **ALLVOLS QUERY**

will be returned if the command was issued with a volume name of '\*'.

#### **MULTIVOL QUERY**

will be returned if the command was issued with more than one volume name.

#### **USEDext**

USEDEXT operands

```
GROUP  REGION  VOLUME  START  SIZE  (END)  OWNER  ADDR  SA  NOTES
-----
ggggggg rrrrrrrr sssssss nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn uuuuuuuu aaaa ssssssss notes
ggggggg rrrrrrrr sssssss nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn uuuuuuuu aaaa ssssssss notes
```

Unless the RETVIA/RETURNVIA prefix keyword is used, the result will be a reader file in the above format. There will be one header line per request. There will usually be one detail line per minidisk entry. The "GROUP" column will show an "\*" for a REGION or VOLUME report. The "REGION" column will show an "\*" for a VOLUME report. The (END) column is calculated as (START+SIZE)-1. If the minidisk is contained within multiple overlapping regions, or in a region that is included in more than one group, there may be multiple entries for that minidisk record.

#### **FREEExt**

FREEXT operands

```
GROUP  REGION  VOLUME  START  SIZE  (END)  OWNER  ADDR  SA
-----
ggggggg rrrrrrrr sssssss nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn .FREE. 0000 *
ggggggg rrrrrrrr sssssss nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn .FREE. 0000 *
```

Unless the RETVIA/RETURNVIA prefix keyword is used, the result will be a reader file in the above format. There will be one header line per query request. There will usually be one detail line per free space entry.

The "GROUP" column will be an "\*" for a REGION or VOLUME report. The "REGION" column will be an "\*" for a VOLUME report. The (END) column is calculated as (START+SIZE)-1.

Note that output is similar to the USEDext report, except for the following:

- OWNER will be ".FREE."
- ADDR will be "0000"
- SYSAFFIN will be "\*"

If the free space is contained within multiple overlapping regions, or in a region that is included in more than one group, there may be multiple entries for that free space record.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. When adding a VOLUME, the *size* value must be greater than the *start* value.
3. The DASD ADD GROUP *group\_name* command updates the :GROUPS. section in the EXTENT CONTROL file. The format is:

```
group_name region_name1 region_name2 ... region_namen
```

4. The DASD ADD REGION *region\_name* command updates the :REGIONS. section in the EXTENT CONTROL file. The format is:

```
regionid volser regstart regend ttttmmm
```

where *regend* is calculated using the *size* and *start* values as follows:

```
regend = (start + size) - 1
```

5. The DASD ADD VOLUME *valid* command updates the :REGIONS. section in the EXTENT CONTROL file. The format is:

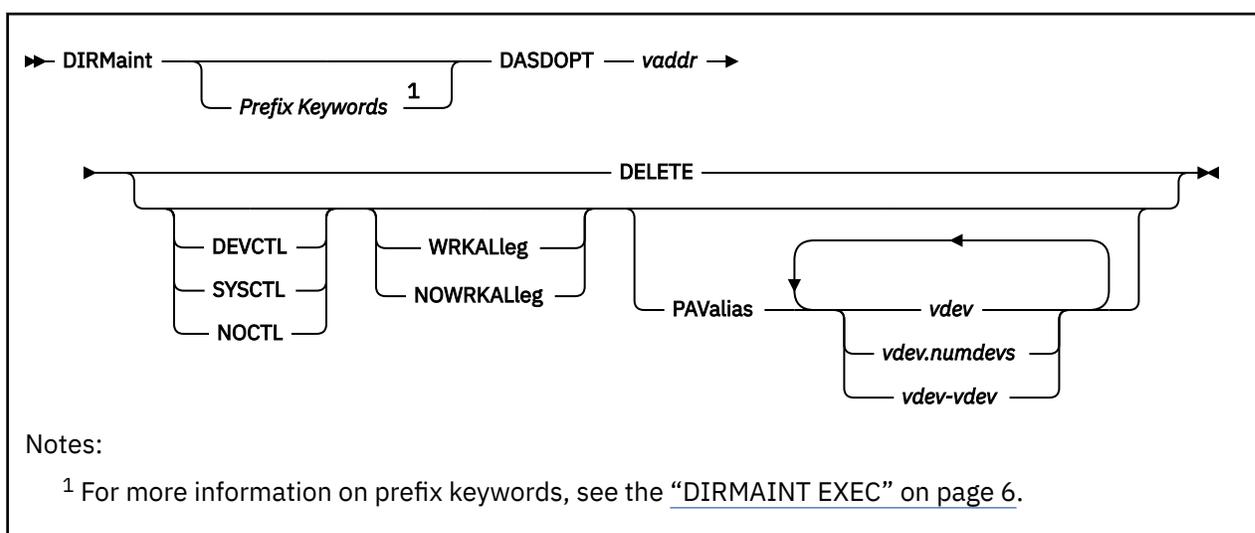
```
volser volser volstart volend ttttmmm
```

where *volend* is calculated using the *size* value as follows:

```
volend = size - 1
```

6. The "Special User IDs in the Default User Directory (USER DIRECT)" section in *z/VM: Installation Guide* recommends using and/or updating special dummy user IDs in the default user directory (USER DIRECT) to track allocation of CP areas on a volume. For example, the \$DIRECT\$ user ID tracks directory cylinders/blocks. The CP QUERY ALLOC MAP command identifies CP reserved areas, and these areas are tracked by DirMaint in the command output as .DRCT., .PAGE., .SPOOL., and .TDISK. areas. The DASD USEDEXT and DASD FREEXT commands do not list overlaps between a special user ID and the allocation extents information when the extents exactly match. This avoids an OVERLAP warning on those extents.

## DASDOPT



### Purpose

Use the DASDOPT operand of the DIRMAINT command to add, change, or delete an existing DASDOPT statement.

### Operands

#### **vaddr**

is the virtual address of the MDISK, DEDICATE, or LINK statement that the new or changed DASDOPT statement should apply.

#### **DELETE**

means that the specified DASDOPT directory control statement should be deleted.

#### **DEVCTL**

means that CCWs that have an effect on system resources and functions directly related to the device will be accepted.

#### **SYSCTL**

means that CCWs that can have a direct global effect on subsystem resources and functions, not just those pertaining to the device, will be accepted.

#### **NOCTL**

means that CCWs that exercise or control subsystem resources or functions will not be accepted.

#### **WRKALleg**

#### **NOWRKALleg**

specifies that working allegiance is active or is not active on the minidisk.

#### **PAValias**

defines one or more alias parallel access volumes for the fullpack minidisk base parallel access volume specified in *vaddr*. One virtual device is created for each specified *vdev*.

**Note:** The PAVALIAS operand is not valid for DEDICATED devices.

#### **vdev**

#### **vdev.numdevs**

#### **vdev-vdev**

specifies a single virtual device (*vdev*), a virtual device combined with a decimal range count (*vdev.numdevs*), or a range of virtual devices (*vdev-vdev*) to be added or updated on the DASDOPT statement.

**Note:**

1. The device numbers entered must be hexadecimal numbers between X'0000' and X'FFFF'.
2. *numdevs* must be a decimal number.
3. Multiple device specifications must not overlap.

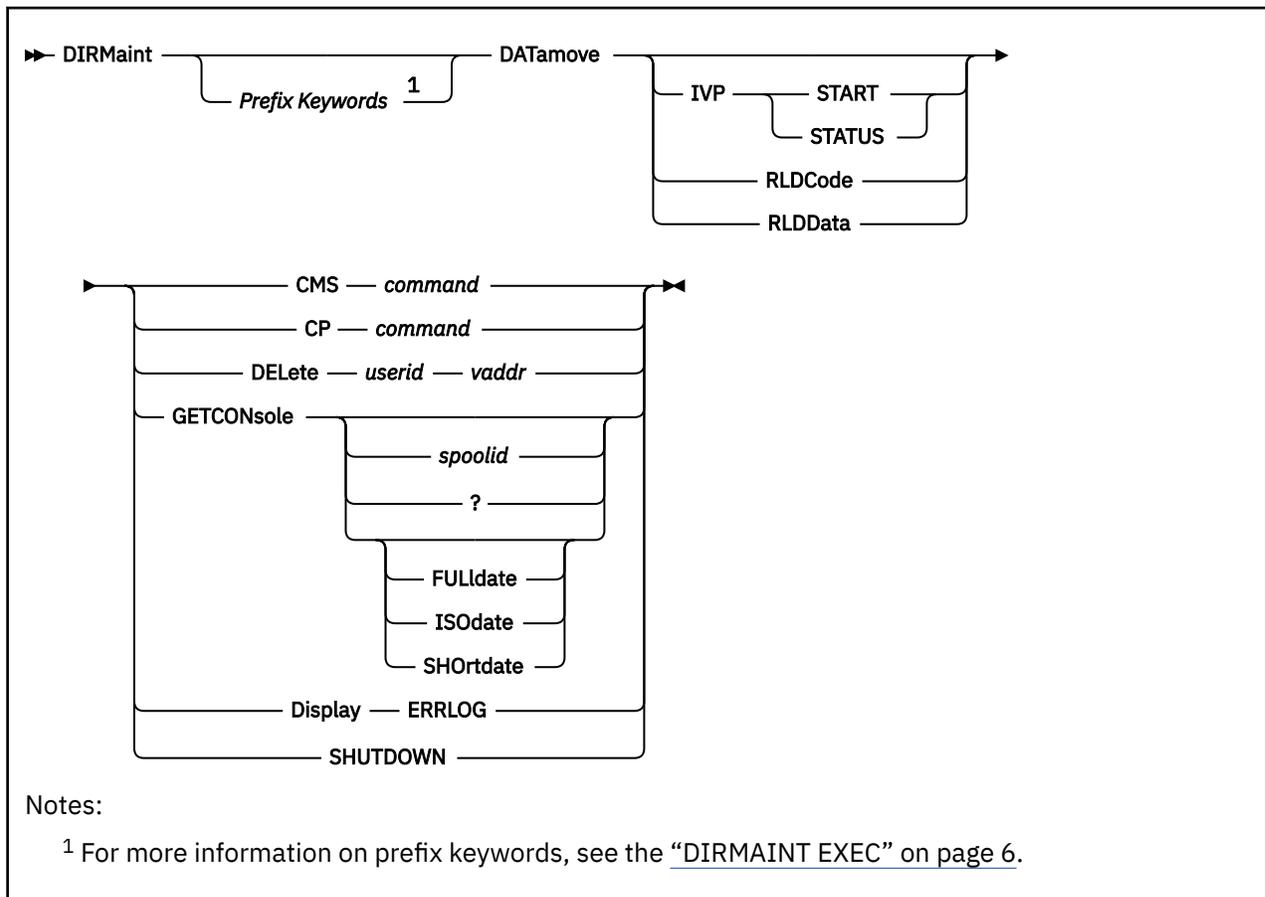
**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the DASDOPT directory statement see *z/VM: CP Planning and Administration*.
3. A DASDOPT statement is a logical extension to a DEDICATE, MDISK, or LINK statement in the directory. It is used to specify the extent of control a virtual machine has over the caching function of a full-volume minidisk or a dedicated device that resides on a cache control unit. DASDOPT statements should only be used with minidisks that are full-volume minidisks.

**Note:** DirMaint does not enforce the full-volume minidisk requirement.

4. Do not place comments or blanks between an MDISK, DEDICATE, or LINK statement and the associated DASDOPT statement if it is manually added to a directory definition.

## DATAMOVE



### Purpose

Use the DATAMOVE operand of the DIRMAINT command to cause DirMaint to direct the DATAMOVE virtual machine to perform the specified function.

When you specify this command without the FOR and AT prefix keyword options, the command is sent to each of the DATAMOVE servers. Use the FOR prefix to specify the name of a specific server to receive this command. Use both the FOR and AT prefixes to specify the name and node of a DATAMOVE server that is running on a different z/VM node from where the DIRMAINT server is running.

### Operands

#### IVP

starts an automated installation verification program run on the specified DATAMOVE machine, or obtains status information on the run on the specified DATAMOVE machine.

#### START

starts an IVP run. A spool file containing the results of the run is sent to the user that issues the IVP START command. See [“IVP Results”](#) on page 139.

#### STATUS

requests status of the most recent IVP run. A spool file containing the status of the run is sent to the user ID that issues the IVP STATUS command. See [“IVP Results”](#) on page 139.

#### RLDCode

causes the DATAMOVE virtual machine to reload its resident operating procedures.

**RLDData**

causes the DATAMOVE virtual machine to reload the data tables from the CONFIG\* DATADVH disk files.

**CP command**

**CMS command**

tells the DATAMOVE service machine to execute the specified CP or CMS command.

**DELeTe *userid vaddr***

specifies that the DATAMOVE virtual machine is to delete the entry specified as *userid vaddr* from the retry queue file.

**Display ERRLOG**

specifies that a copy of the DATAMOVE machine's retry queue file is to be sent to the requestor's virtual reader as DVHDMX ERRLOG.

**GETCONsole *spoolid***

specifies that a copy of the identified console spool file is to be sent to the command issuer. If *spoolid* is omitted, the currently active console spool file is closed and a copy is sent to the command issuer.

**Note:** Specifying the spool ID of the currently active console spool file will result in an error. To obtain a copy of the currently active console spool file, omit the *spoolid* parameter.

**GETCONsole ?**

specifies that a list of the available console spool files is displayed to the command issuer.

**FULLdate**

Uses the FULLDATE option on the CP QUERY PRT command.

**ISODate**

Uses the ISODATE option on the CP QUERY PRT command.

**SHORtdate**

uses the SHORTDATE option on the CP QUERY PRT command.

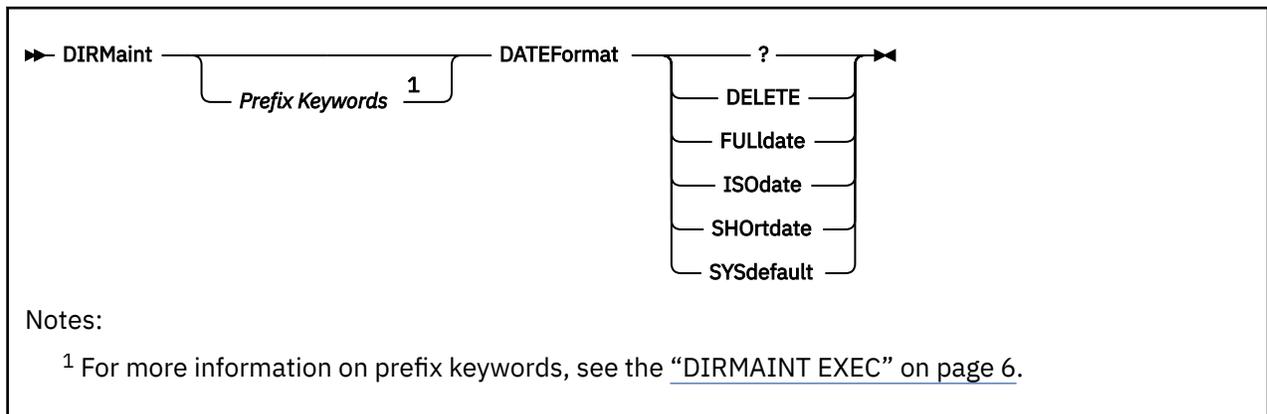
**SHUTDOWN**

specifies that the DATAMOVE virtual machine session is to be terminated.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. The DATAMOVE SHUTDOWN operand is queued on the DATAMOVE virtual machine command queue and will be processed in sequence. Any commands remaining on the queue after the SHUTDOWN operand will not be processed.
3. The DVHDMX ERRLOG file is a copy of the DATAMOVE machine's retry queue file created when requested by the DIRM DATAMOVE DISPLAY ERRLOG command. The retry queue file contains a record of work to be retried because a CP LINK command failed to obtain access to a disk. The retry will take place the next time the object directory is updated. Other errors encountered while processing a FORMAT, COPY, or CLEAN request are *rolled back*, with the user's original disk being returned to the user and no record of the failure in the retry queue on the DATAMOVE machine. The console spool file and optional TRANSLOG file contains this historical information. The DIRMAINT machine keeps a record of failed work units. This includes errors encountered in the DATAMOVE machine processing. This information can be displayed using the DIRM STATUS WORKUNIT FAILED command.
4. For more information about the FULLDATE, ISODATE, or SHORTDATE operands, type HELP CP QUERY or see [z/VM: CP Commands and Utilities Reference](#).

## DATEFORM



### Purpose

Use the DATEFORMAT operand of the DIRMAINT command to allow the general user to add, change, delete, or query the DATEFORMAT statement in a directory entry.

### Operands

?

issues a query against the existing DATEFORMAT statement.

#### DELETE

deletes an existing DATEFORMAT statement from the directory entry.

#### FULLdate

requests that dates be shown in FULL date, *dd/yyyy*, format.

#### ISOdate

requests that dates be shown in ISO date, *yyyy-mm-dd*, format.

#### SHOrtdate

requests that dates be shown in SHORT date, *mm/dd/yy*, *mm/dd*, or *dd mmm yy* format depending on the particular command.

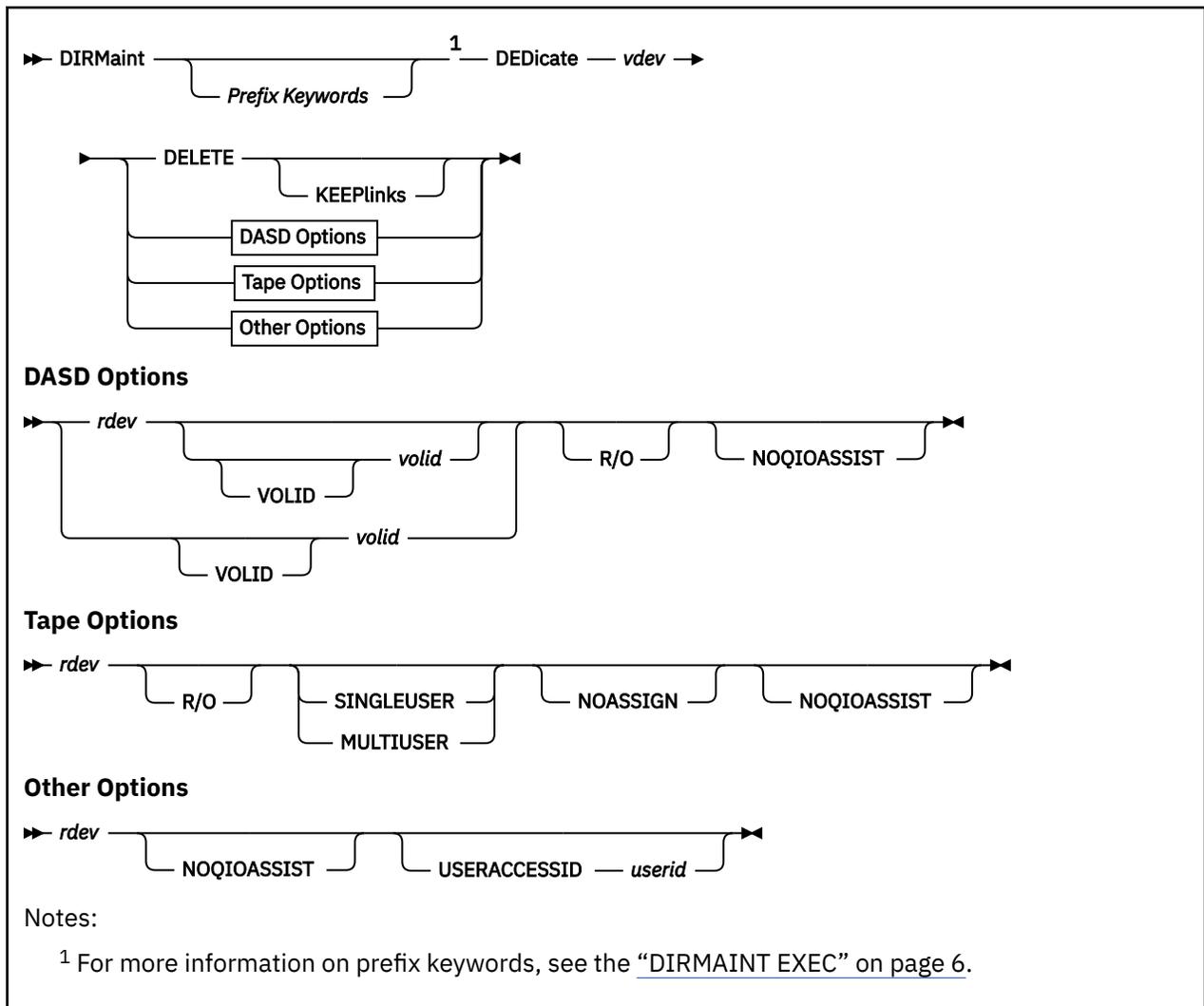
#### SYSdefault

requests that dates be shown in whichever format has been selected by the system administrator and specified with the SYSTEM\_DATEFORMAT statement in the system configuration file, or by the system operator and specified with a CP SET DATEFORMAT SYSTEM command.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the DATEFORMAT directory statement, see [z/VM: CP Planning and Administration](#).

## DEDICATE



### Purpose

Use the DEDICATE operand of the DIRMAINT command to add or delete an existing DEDICATE statement.

### Operands

#### **vdev**

is the virtual address associated with the DEDICATE statement.

#### **DELETE**

means that the specified DEDICATE directory control statement should be deleted.

#### **KEELinks**

specifies that the links to the deleted volume are to be kept. This allows for replacement of the deleted volume without having to reestablish the links.

#### **rdev**

is the real device number.

#### **VOLID**

specifies that the value following it is a volume label.

**valid**

is the volume label of a disk pack mounted on a real disk storage device. It must be 1 to 6 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +. If *valid* is less than 5 characters, the VOLID keyword is required to distinguish it from an *rdev* address.

**R/O**

specifies that the virtual device is to be in read-only mode.

**SINGLEUSER**

specifies a tape device to a single user.

**MULTIUSER**

specifies attach a tape device to multiple users so that the device can be serially shared.

This operand must be specified on the dedicate statement by all users who intend on sharing the device. If a user logs on and has a dedicate statement for a device without this operand, the device becomes dedicated to that user and can not be attached MULTIUSER by anyone until the device is detached. This operand is valid only for 3480, 3490, and 3590 tape devices.

**NOASSIGN**

indicates that the attach process should not issue an assign channel command for this user.

**NOQIOASSIST**

indicates the device is not eligible for Queued-I/O Assist.

**USERACCESSID *userid***

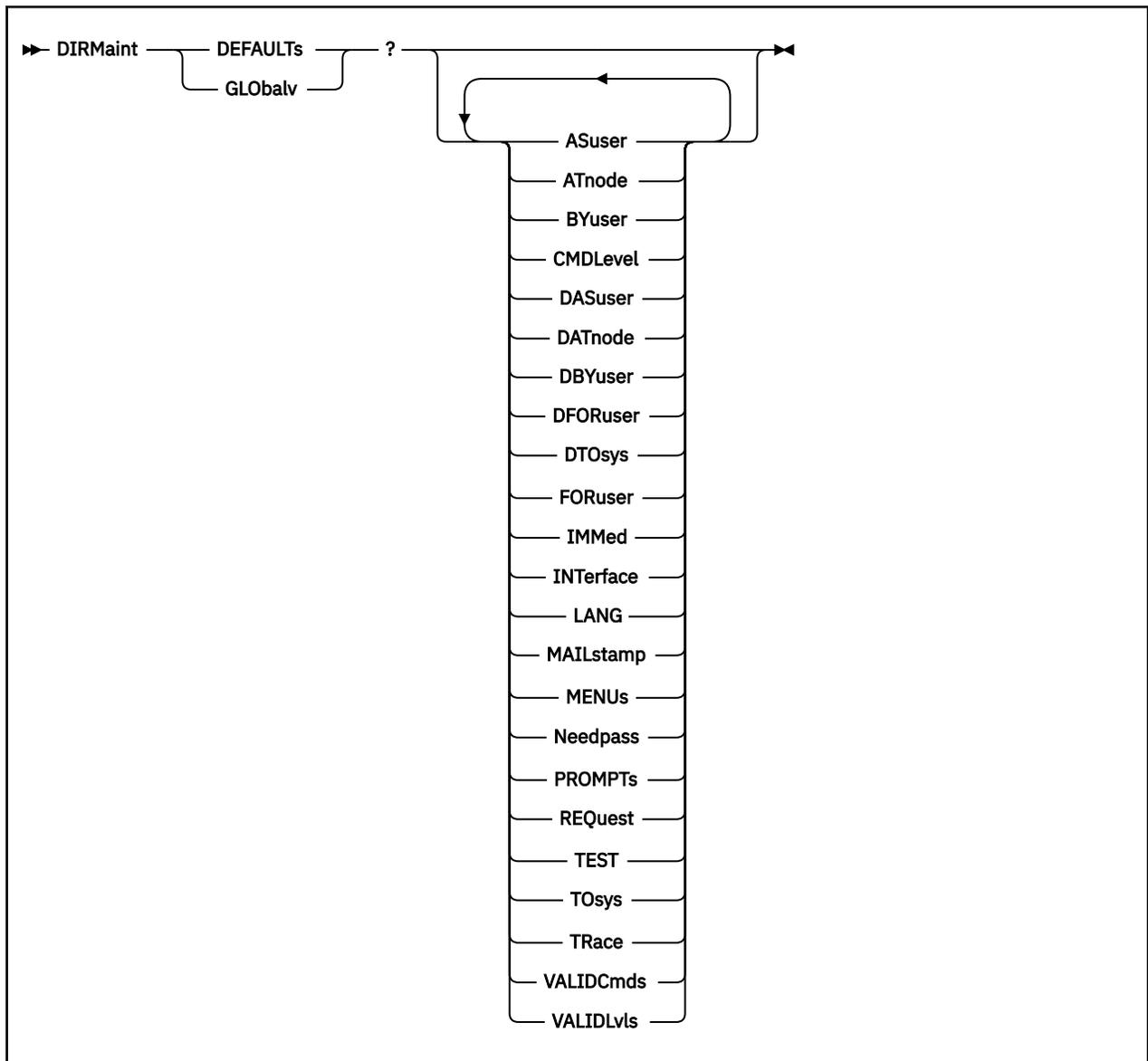
indicates that FCP logical unit number (LUN) access should be given to the specified *userid*. z/VM will store the *userid* into the subchannel, and the adapter will use these stored names to correlate I/O requests with the rules in the access control table (ACT) and allow access to the LUNs based on those rules.

**Usage Notes**

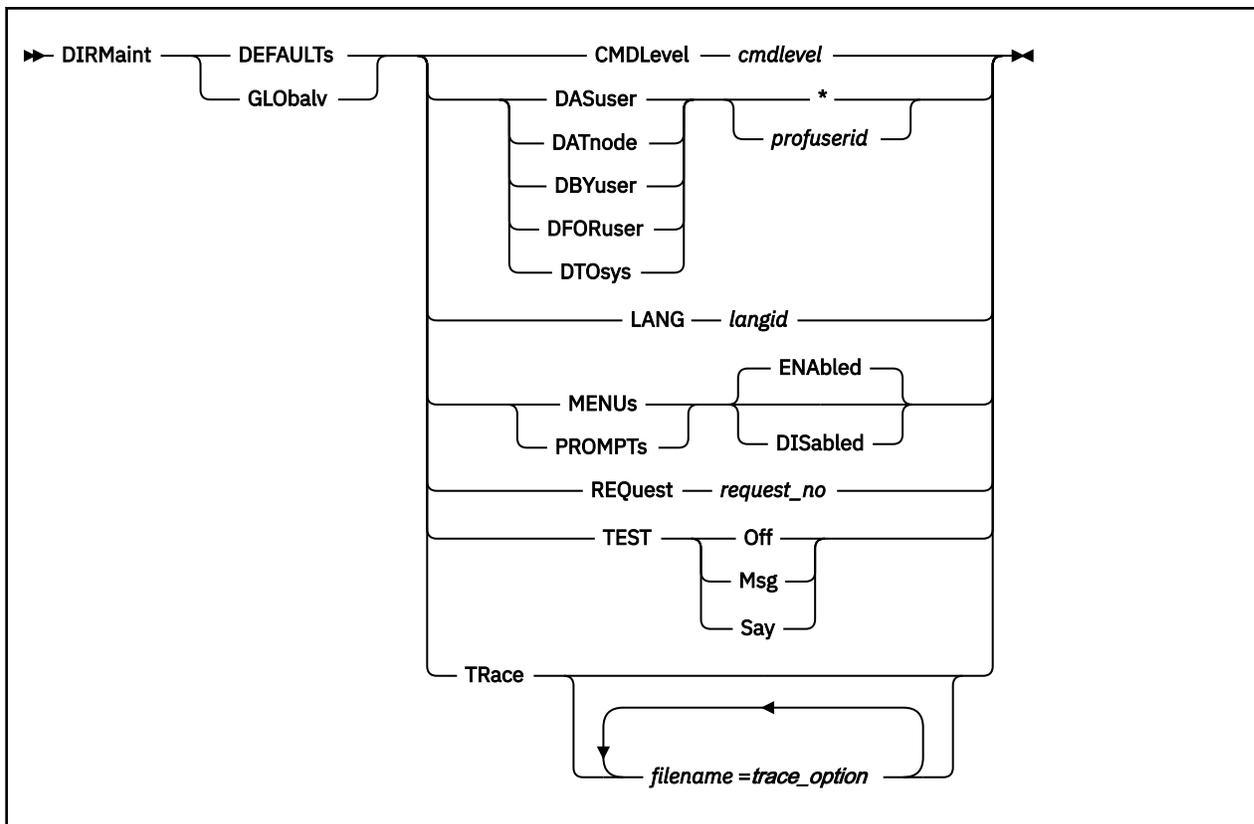
1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the DEDICATE directory statement see *z/VM: CP Planning and Administration*.
3. If more than one virtual machine has a dedicate statement for a given real device, only the first virtual machine to log on receives control of the device.
4. All options are not available on all releases of VM.

## DEFAULTS, GLOBALV

Use this command syntax to determine the existing value of a DirMaint GLOBALV variable.



Use this command syntax to set the value of a DirMaint GLOBALV variable.



### Purpose

Use the DEFAULTS or GLObalv operand of the DIRMAINT command to set or query various default values used within DirMaint processing.

### Operands

?

Causes the current value of one or more variables to be displayed.

#### ASuser

Is set by the program from the ASUSER prefix keyword on the DIRM command (if specified), or from the DASUSER global variable (if previously set), or to an \* by default.

#### ATnode

Is set by the program from the ATNODE prefix keyword on the DIRM command (if specified), or from the DATNODE global variable (if previously set), or to an \* by default.

#### BYuser

Is set by the program from the BYUSER prefix keyword on the DIRM command (if specified), or from the DBYUSER global variable (if previously set), or to an asterisk (\*) by default.

#### CMDLevel *cmdlevel*

May be set by the user to indicate the compatibility level desired for use with application programs. The valid values are 150A or 140A.

#### DASuser

Identifies the default value for the ASUSER prefix keyword on the DIRMAINT command. The valid values are an asterisk or a valid file name which may be used as a user ID.

#### DATnode

Identifies the default value for the ATNODE prefix keyword on the DIRM command. The valid values are an asterisk or a valid file name which may be used as a network node ID.

## DEFAULTS

### **DBYuser**

Identifies the default value for the BYUSER prefix keyword on the DIRM command. The valid values are an asterisk or a valid file name which may be used as a user ID.

### **DFORuser**

Identifies the default value for the FORUSER prefix keyword on the DIRM command. The valid values are an asterisk or a valid file name which may be used as a user ID.

### **DTOsys**

Identifies the default value for the TOSYS prefix keyword on the DIRM command. The valid values are an asterisk or a valid file name which may be used as a network node ID.

### **IMMed**

Is set by the program based on the presence or absence of the IMMED keyword prefix on the DIRM command.

### **LANG *langid***

A valid *langid* consists of 5 characters or less, that are valid in a CMS file name, and is explicitly defined on one or more *langid* USER MSGS entries in the CONFIG\* DATADVH file(s).

Also a *langid* of *1SAPI* may be used to specify the *application programming interface* language, where all replies are:

```
REQUEST= nnnn RTN=DVHXXX MSGNO=1234 FMT=01 SUBS= VAR1 VAR2
```

### **MAILstamp**

Displays the last time the mail file was sent to you.

## **MENUs**

### **ENabled**

specifies that a “fill in the blanks” menu will be displayed when required operands are not provided on a command invocation. This is the default.

### **DISabled**

prevents the display of the operand menu.

## **Needpass**

The Needpass value is ignored and password prompting is not done for DirMaint commands when the NEEDPASS\_RESPECT configuration option is either allowed to default to NO or is specifically set to NO. When the NEEDPASS\_RESPECT configuration option is set to YES, the Needpass value indicates whether password prompting is done for most DirMaint commands (YES) or is not done (NO).

## **PROMPTs**

### **ENabled**

specifies that a prompt will be issued when a required password is not provided on a command invocation. This is the default.

### **DISabled**

prevents the prompt for the password.

## **REQuest *request\_no***

Specifies an identifying number to be associated with a transaction sent to the DIRMAINT server. Used by programs using LANG 1SAPI to associate the transactions with the returned messages.

## **TEST**

Determines whether or not a DIRM command results in a transaction to the DIRMAINT service machine (TEST OFF), or is displayed on the invoker's console as a CP message (TEST MSG), or is displayed on the invoker's console using a CMS REXX SAY statement (TEST SAY). The valid values are OFF, MSG, or SAY.

## **TOSys**

Is set by the program from the TOSYS prefix keyword on the DIRM command (if specified), or from the DTOSYS global variable (if previously set), or to an \* by default.

**TRace**

Is used for problem diagnosis to activate REXX tracing for the listed routines, and specify the degree of detail desired in the trace. Each entry must be in the form of a file name, followed by an equal sign, followed by a REXX trace option (All, Commands, Error, Failure, Intermediates, Labels, Normal, Off, Results, Scan); with no intervening blanks.

**VALIDCmds**

Is set by the program to identify the valid values for CMDLEVELS. The valid values are 150A and 140A.

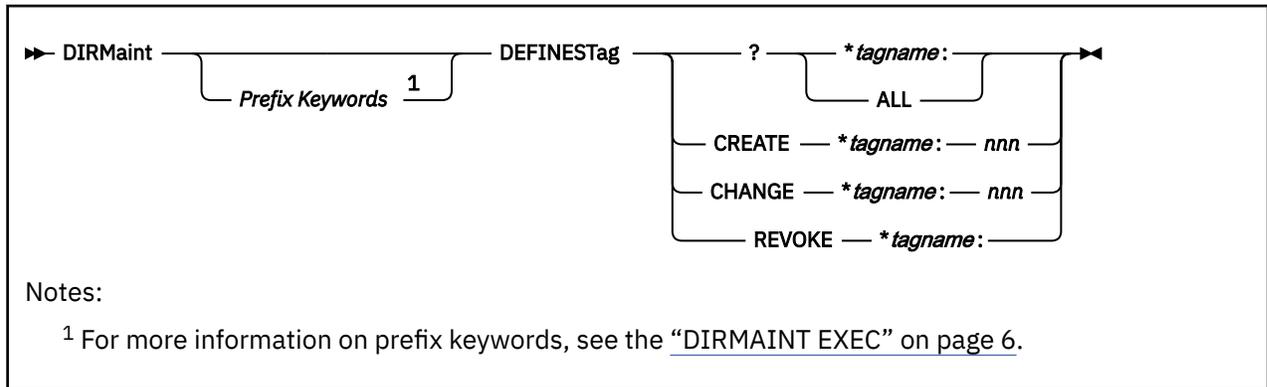
**VALIDLvl**

Is set by the program to identify the valid values for the interface level. The current value is 199501.

**Usage Notes**

1. DirMaint prefix operands do not apply to this command. They are accepted without error, but otherwise ignored.
2. The DEFAULTS and GLOBALV commands are completely handled within the virtual machine of the command issuer. No transaction is sent to the DIRMAINT service machine, therefore no password is required or requested for authentication, even if you have set NEEDPASS to YES.
3. The DEFAULTS and GLOBALV commands are not valid within a BATCH command file.
4. For more information on altering the NEEDPASS setting, see [“NEEDPASS” on page 171](#).
5. All global variables are stored in the DVH15 variable pool.

## DEFINESTAG



### Purpose

Use the DEFINESTAG operand of the DIRMAINT command to manipulate user defined tagged comments. An installation can define local tags that can be stored in the CP directory and manipulated by DirMaint. This may be useful for information normally placed into comments, such as department information. This command creates the required definitions within the DIRMAINT machine. It is not used to assign data to a local tag, it is only used to create and manipulate a local tag.

### Operands

**?**

requests that the current setting of the specified tag (or all tags, using the ALL keyword) be returned in a message.

**\*tagname:**

is a user defined tag name. The following rules apply:

- It must start with an \* and end with a :
- The body of the tag (between the \* and the :) must be 10 characters or less.
- The body of the tag is restricted to alphanumeric characters.

**ALL**

requests that all tags be reported in the query.

**CREATE**

attempts to create a new tag.

**CHANGE**

attempts to alter an existing tag definition's sort sequence.

**nnn**

specifies the sort order of the created tag in the range of 000 to 999.

**REVOKE**

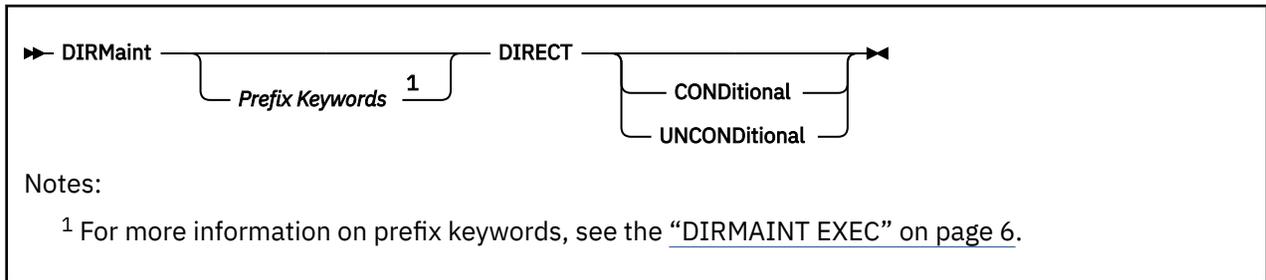
requests that the specified tag be removed from DirMaint's internal tables. The tag can no longer be set or queried by users.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.

2. The sort value (*nnn*) used with CREATE and CHANGE assigns a unique sort order to each user defined tag. Internally, this sort order is *8nnn*. The *nnn* is used to assign a relative sorting order in regards to user defined tags only. It does not affect the sorting order of CP directory statements.
3. Currently the number of user defined tags is limited to 1000 (using sort orders 000 to 999).
4. When using the CREATE keyword, the tag being created must not already exist and the provided sort order code must not be in use by any other tag.
5. When using the CHANGE keyword, the tag name being changed must already exist and the new sort order must not already be in use.
6. When using the REVOKE keyword, DirMaint makes sure that the specified tag is not being used in any directory. If it is, the request will fail. If you wish to disable this tag and do not wish to remove it from all user directories, the file containing the tag definition information (LOCAL DATADVH) can be altered on the DIRMAINT server. It is located on the primary directory disk. Once an entry is removed from this file, the tags in the user directories no longer have special meaning to DirMaint and are treated like comments.
7. The widespread use of local tags has the ability of significantly expanding the size of the source directory. This should be kept in mind when defining local tags and authorizing users to use them.

## DIRECT



### Purpose

Use the DIRECT operand of the DIRMAINT command to place the current directory structure online.

### Operands

#### CONDitional

specifies that the DIRECTXA command will not be issued unless there are pending changes to be processed.

**Note:** DirMaint does not keep track of pending changes to the object directory when updates to the object directory are disabled. To update the object directory when object directory updates are disabled, use DIRM DIRECT UNCONDITIONAL.

#### UNCONDitional

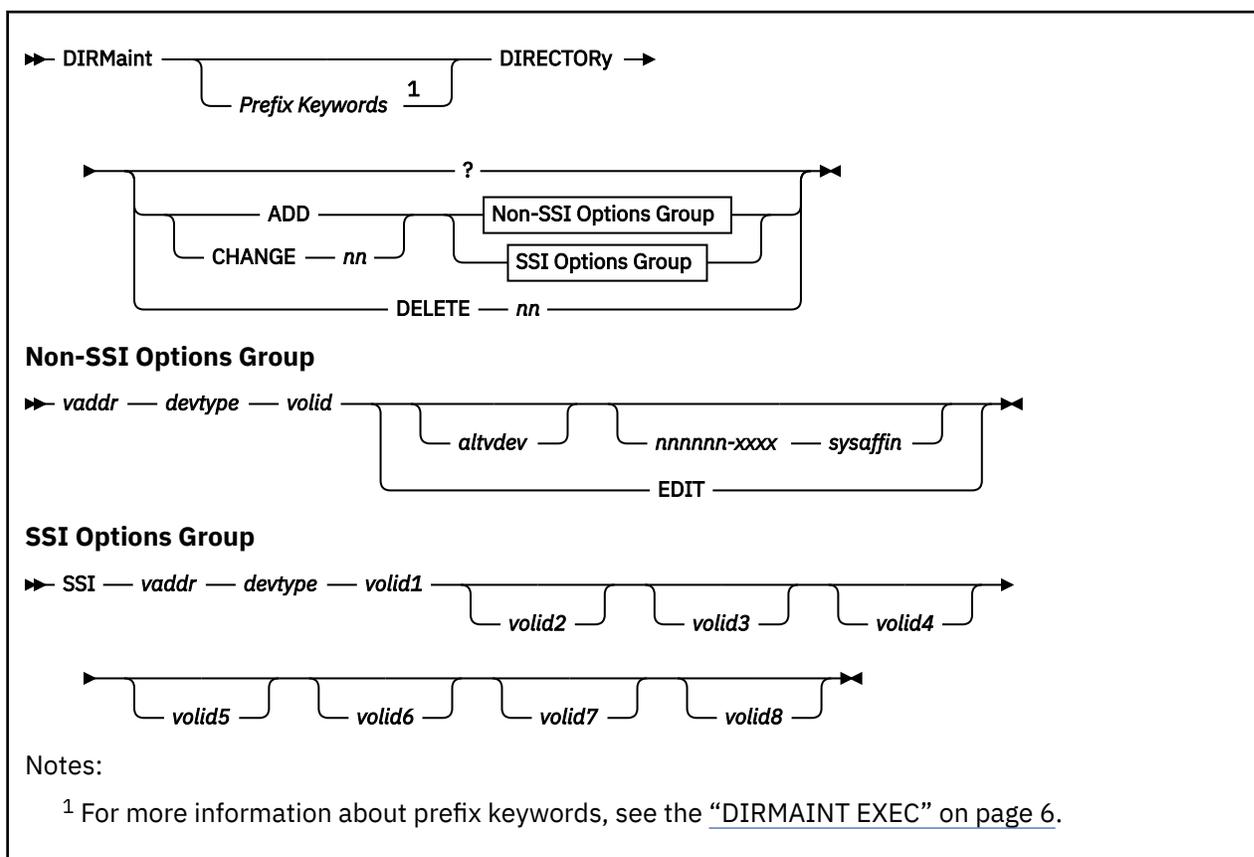
specifies that the DIRECTXA command will be issued, regardless of any pending changes to be processed.

**Note:** If neither the CONDitional nor UNCONDitional keyword is specified, then the default will be determined by the DEFAULT\_DIRECT\_ACTION=CONDITIONAL|UNCONDITIONAL record in the CONFIG\* DATADVH file. If this record is not specified, or is invalid, then the default will be UNCONDITIONAL.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOSys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. DIRECTXA places the directory online.
3. For more information on the DIRECTXA command, see *z/VM: CP Commands and Utilities Reference*.
4. Any errors reported by the VM directory program are reflected in the DirMaint messages.
5. Should an error result during the operation of this command (issued from the CP DIRECTXA module), DirMaint will disable source directory updates automatically. The source directory can be enabled for updates using the DIRM ENABLE command.

# DIRECTORY



## Purpose

Use the DIRECTORY operand of the DIRMAINT command to add, change, or delete the DIRECTORY directory control statement. The DIRECTORY statement defines to CP the device on which space has been allocated for the directory.

## Operands

?

returns the settings of the current DIRECTORY directory control statement specifications. The directory statements listed in the response are numbered from 1 to *n*. The position is specified with the CHANGE and DELETE operands, to indicate which statement is being changed or modified.

### ADD

specifies that a DIRECTORY directory control statement containing the specified information is to be added to the system directory.

### CHANGE *nn*

requests that the provided information replace the specified existing directory statement. The *nn* value specifies the number of the statement that is to be modified. The statement number can be obtained by issuing the DIRMAINT DIRECTORY ? command.

### DELETE *nn*

requests that the specified existing directory statement (the *nn*th directory statement) be deleted. The *nn* value specifies the number of the statement that is to be deleted. The statement number can be obtained by issuing the DIRMAINT DIRECTORY ? command.

### *vaddr*

specifies the virtual device number of the device that is used to contain the object directory.

## DIRECTORY

### ***devtype***

specifies the device type. The valid device types are:

- FB-512
- 3380
- 3390
- 9336

### ***valid***

specifies the volume label of the directory volume. The variable *valid* must be 1 to 6 characters from the following list: A-Z, 0-9, #, @, \$, :, -, and +.

**Note:** Up to 8 *valids* can be specified when the SSI keyword is also specified.

### ***altvdev***

specifies an alternative virtual device number, on which to write the directory if the primary virtual device number is unavailable.

### ***nnnnnn-xxxx***

specifies the processor ID of the system to which the DIRECTORY statement applies. If the same *nodeid* is used for all CPUs in an *n*-way processor, the first character of the serial number can be specified as an asterisk (*\*nnnnn-xxxx*, for example).

### ***sysaffin***

specifies a 1- to 8-character alphanumeric string that identifies the system whose object directory is affected by the SYSAFFIN statements.

### **EDIT**

defines a special work volume to be used by the DIRECT command when it is entered with the EDIT option. You can only specify the EDIT option on one DIRECTORY statement, and it must be the last of the set of DIRECTORY statements. DIRECTXA validates the syntax of this statement, but ignores its contents.

### **SSI**

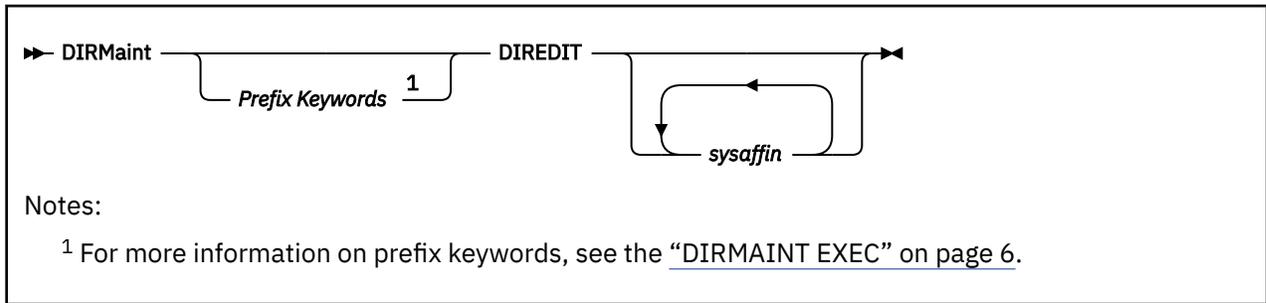
specifies that this directory is to be shared among all systems in a single system image (SSI) cluster. Note that the *altvdev*, *nnnnnn-xxxx*, *sysaffin*, and EDIT operands are not allowed when SSI is specified.

**Note:** Up to 8 *valids* can be specified when the SSI keyword is also specified.

## Usage Notes

1. You can use one or more of the prefix keywords (ASuser, BYuser, NODIRECTXA, PRESET, REQUEST, TOSYS) with this command. These prefix keywords can be specified in any order, but each keyword can only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. When an SSI DIRECTORY statement is added or changed and the resulting statement is larger than can be contained in a single (70-character) statement, DirMaint will split the statement into two statements, with the second statement added to the end of the list of DIRECTORY statements.
3. For more information about the DIRECTORY directory statement, see [z/VM: CP Planning and Administration](#).

## DIREEDIT



### Purpose

Use the DIREEDIT operand of the DIRMAINT command to syntactically check the current directory structure through the use of the EDIT feature of the VM directory program.

### Operands

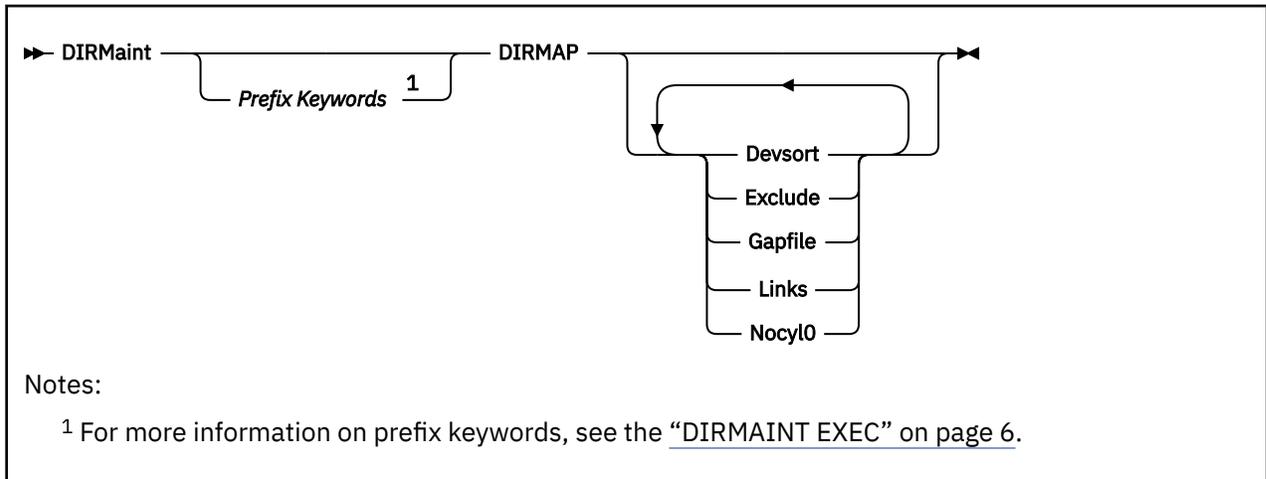
#### *sysaffin*

specifies that the edit operation should pass the *sysaffin* value to the DIRECTXA module.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, MENU) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The directory is not placed online, it is only checked for syntax.
4. DIRECTXA edit checks the directory structure.
5. For more information on the DIRECTXA command, see [z/VM: CP Commands and Utilities Reference](#).
6. Any errors reported by the z/VM directory program are reflected in the DirMaint messages.

## DIRMAP



### Purpose

Use the DIRMAP operand of the DIRMAINT command to provide a report detailing the current DASD utilization on the system.

### Operands

#### Devsort

requests that the report is to be sorted by device type followed by volume label. The default is to sort by volume label.

#### Exclude

specifies that EXCLUDED extents are not to be considered when generating reports.

#### Gapfile

Generate a file listing gaps in addition to a report detailing the current DASD utilization.

**Note:** If you use the GAPFILE option without using the EXCLUDE option and excluded full volume overlays are defined on your system no gap data will be returned. Use the EXCLUDE option to eliminate the excluded full volume overlays from consideration when building the gap data.

The gapfile is listed in the following format:

#### Field

##### Use

- 1** valid of disk
- 2** device type of disk.
- 3** start cylinder/block of gap
- 4** end cylinder/block of gap
- 5** number of cylinders/blocks in gap

**Note:** All fields are blank delimited and not bound to a specific column range.

**Links**

requests that a LINKMAP be produced. This file contains a map of all LINKs to MDISKs on the processed volumes. The default is to only generate a DISKMAP.

**Nocyl0**

Forces records that are destined for the GAPFILE to be recomputed to use cylinder 1 as a start to prevent a gap being identified as starting on cylinder 0. This option is only meaningful to GAPFILE output. The Mdisk-map shows cylinder 0 as a gap (if it is not used) regardless of this option.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, MENU, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. Since DIRMAINT only maintains active volume control files for non-full volume minidisks, the DASD map reports generated by this command do not include any full volume minidisk entries.

## DISABLE



Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the DISABLE operand of the DIRMAINT command to cause incoming directory updates to be rejected.

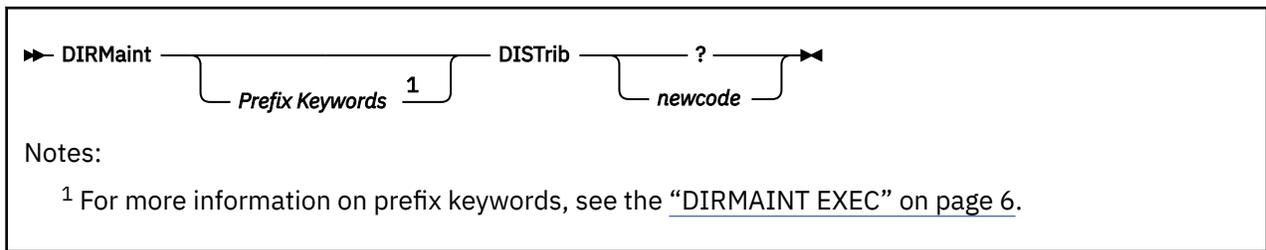
### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Use this operand to temporarily suspend updates to the source directory. When this operand is used, read-only requests will still be processed.
3. The disable function of this operand is automatically invoked if the DIRECTXA program returns an error when attempting to place the directory online.

**Note:** This only occurs as the result of a DIRM DIRECT command and indicates a problem with the directory structure.

4. Normal operation can be resumed by invoking the DIRMAINT ENABLE operand.

## DISTRIB



### Purpose

Use the DISTRIB operand of the DIRMAINT command to change the distribution code that is printed or punched with your user ID in the separator for spooled output. The distribution code, if defined, exists as a parameter on the ACCOUNT control statement in your directory entry.

### Operands

**?**

allows you to query your current distribution code.

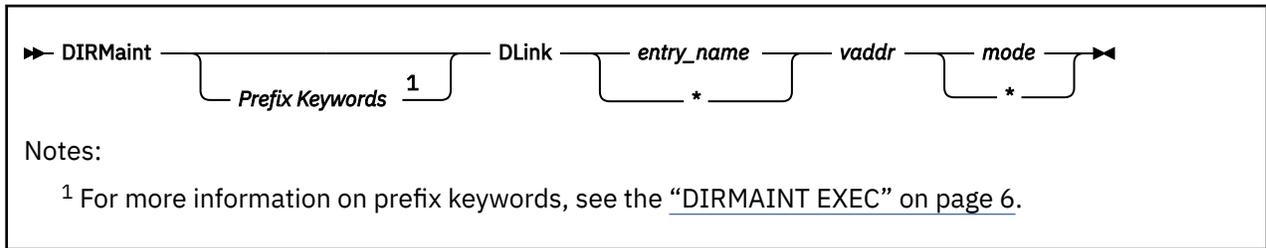
**newcode**

is the new distribution code. This is the value that appears below your user ID on printed output.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Because of the query capability for this operand, a distribution code of ? can not be installed in a user's directory definition using the DISTRIB operand. If such a distribution code is desired, it must be set using the GET and REPLACE operands.
3. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## DLINK



### Purpose

Use the DLINK operand of the DIRMAINT command to remove the LINK statements.

DLINK does not physically alter the source directory. It creates a batch file containing one or more

```
FOR entry_name LINK * addr addr DELETE
```

statements. The batched commands run as low priority background work that removes the LINK statements from the linking user IDs' directory definitions.

### Operands

#### **entry\_name**

specifies the name of the user, identity, or subconfiguration entry whose link to this minidisk is to be dropped. All user links to this minidisk can be dropped by specifying an asterisk (\*) instead of a specific entry name.

#### **vaddr**

specifies the virtual address of the minidisk from which links are to be dropped.

#### **mode**

specifies the type of link to be dropped. All links regardless of the LINK mode can be dropped by specifying an asterisk (\*) instead of a specific MODE. The file modes that can be used are: R, W, M, or \*.

### Usage Notes

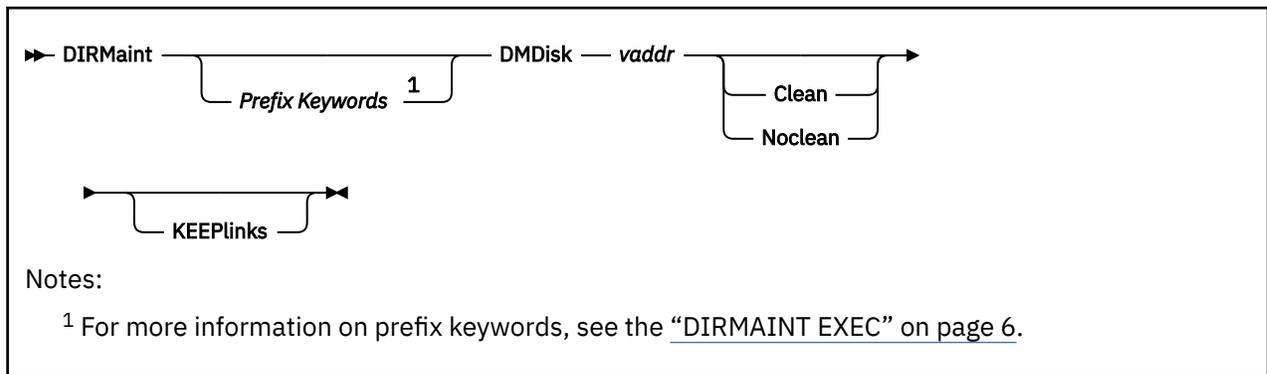
1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. To determine the links that are in effect to your minidisks, issue

```
DIRM REVIEW
```

The returned output will contain a list of all links to your minidisks, identifying the linking user ID and their link mode.

3. The ANYlink and DIRMlink operands are no longer a part of the command syntax. If specified, they are tolerated for compatibility with earlier releases of DirMaint.

## DMDISK



### Purpose

Use the DMDISK operand of the DIRMAINT command to delete a minidisk from a user, identity, or subconfiguration directory entry.

### Operands

#### *vaddr*

specifies the virtual device address of the minidisk.

#### **Clean**

specifies that the released minidisk extent is to be transferred to the user defined as a DATAMOVE machine in the CONFIG\* DATADVH file for formatting, even if the DISK\_CLEANUP= YES option is not in effect (see Usage Note [“3”](#) on page 101).

#### **Noclean**

specifies that the released minidisk extent is to be made available for reassignment immediately without being cleaned, even if the DISK\_CLEANUP= YES option is in effect (see Usage Note [“3”](#) on page 101).

#### **KEEPlinks**

may be specified to suppress the removal of links to the minidisk being deleted. The default is to remove any links to a minidisk being deleted by the DMDISK operand.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. The extent currently occupied by this minidisk will become available for reallocation immediately, unless the installation wants obsolete extents to be formatted before reassignment.
3. If the installation wants released minidisks to be formatted before reassignment, then two CONFIG\* DATADVH file statements are consulted.
  - DISK\_CLEANUP=
  - CYLO\_BLK0\_CLEANUP=

DISK\_CLEANUP determines (in general) whether or not minidisk cleans should take place. CYLO\_BLK0\_CLEANUP is consulted if the extent to be cleaned is using cylinder or block 0. In addition, minidisks will not be cleaned if they form an overlap with another uninvolved extent.

If cleaning is required, the minidisk will be transferred to the user defined as a DATAMOVE machine in the CONFIG\* DATADVH file. A table entry is made denoting that the minidisk is to be formatted. The

old extent will not be available for reassignment until DIRMAINT is notified that the extent has been formatted.

This processing can be controlled on a transaction basis. If CLEANUP is the normal mode of operation, it can be bypassed by the NOCLEAN option. If CLEANUP is not in effect, it can be forced for a given transaction by including the CLEAN option. This overrides the action dictated by the DISK\_CLEANUP= and CYLO\_BLK0\_CLEANUP= configuration file entries.

The CLEAN option cannot be used if the subject *userid* and the DATAMOVE *userid* are the same.

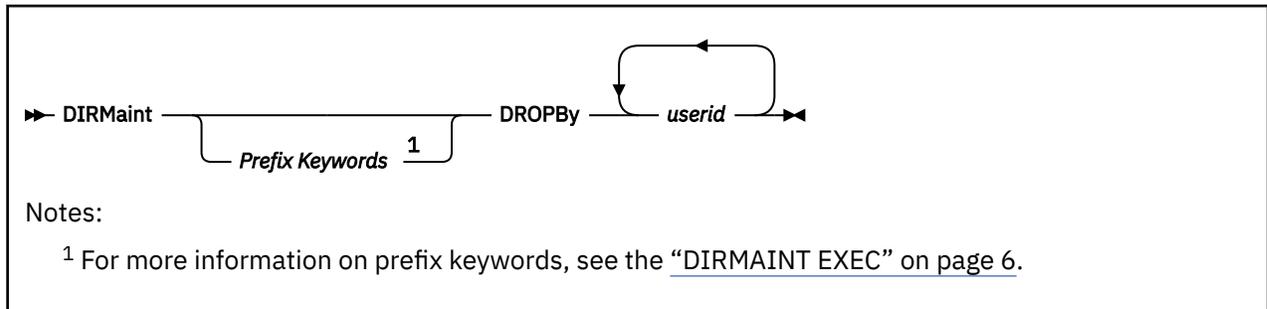
4. DMDISK processing will schedule low priority work to remove all links to the minidisk being deleted unless the KEEPLINKS option is specified.
5. If one of the DATAMOVE virtual machine's minidisks is specified to be cleaned the request will be rejected.
6. Be aware that the minidisk being deleted may be physically linked by other users at the time of its deletion. The DATAMOVE virtual machine postpones the formatting of any disk until it is free of links. It is the installation's responsibility to ensure that these physical CP links are removed. Failure to remove the CP links may result in significant delay before the extents are available for reuse.
7. A DEVNO MDISK may be deleted with the DMDISK command.

**Note:** DISK\_CLEANUP= YES entry in the CONFIG\* DATADVH file and the CLEAN option on the DMDISK command are both ignored for DEVNO MDISKS.

DEVNO MDISKS will not be formatted by DATAMOVE. If there is residual data that must be protected, it is the responsibility of the owning user to format the disk or otherwise erase the data before requesting the administrator to delete the minidisk or purge the user ID.

8. A T-DISK or V-DISK may be deleted using DMDISK. The space for these may or may not be cleaned by the system before reuse. The DISK\_CLEANUP= YES entry and the CLEAN option on the DMDISK command are both ignored for these disks.
9. If a DASD management work unit associated with this command becomes stalled or hung on the DATAMOVE machine, manual intervention may be necessary in order for the work unit to complete. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.
10. If the AT prefix is specified for a subconfiguration entry when the DMDISK operand is issued, the AT keyword value will be used to select the DATAMOVE machine configured at the specified node. In this respect, the FORuser prefix keyword must specify the subconfiguration entry name in order to modify a subconfiguration entry with the DMDISK operand.
11. If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## DROPHY



### Purpose

Use the DROPHY operand of the DIRMAINT command to revoke the authority given by DIRM AUTHBY to utilize the BYuser prefix operand.

### Operands

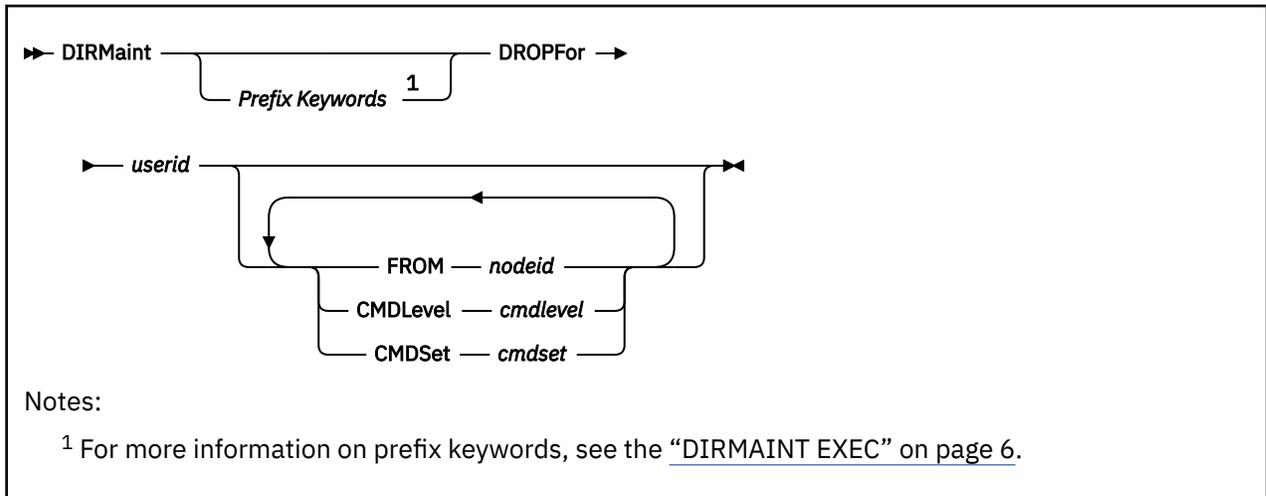
#### *userid*

a list of one or more user IDs whose authorization is revoked.

### Usage Notes

You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.

## DROFFOR



### Purpose

Use the DROFFOR operand of the DIRMAINT command to revoke the authority granted by the AUTHFOR command to issue a command on behalf of another user.

### Operands

#### *userid*

a user ID whose authorization is revoked.

#### **FROM** *nodeid*

specifies the node ID of the user whose authorization is being dropped.

#### **CMDLevel** *cmdlevel*

specifies the command level being dropped.

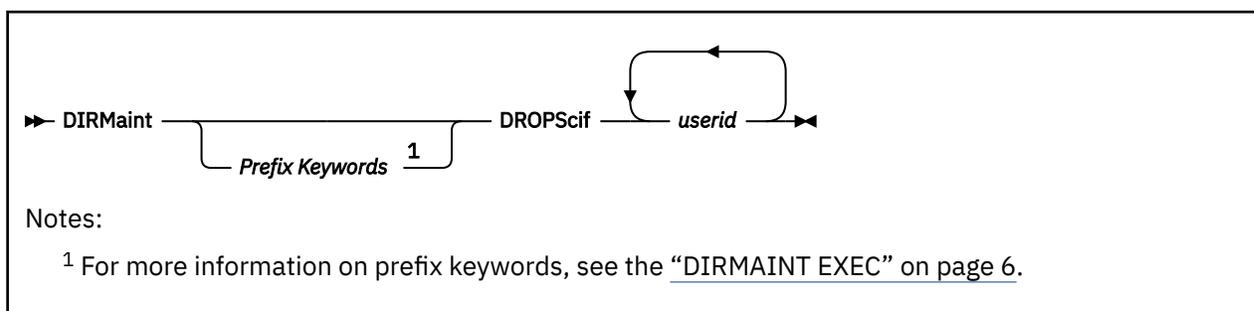
#### **CMDSet** *cmdset*

specifies the command set being dropped.

### Usage Notes

You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.

## DROPSCIF



### Purpose

Use the DROPSCIF operand of the DIRMAINT command to revoke the authority granted by a previous AUTHSCIF command to utilize another user as a secondary console.

### Operands

#### *userid*

is a list of one or more users having their authority revoked.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. The use of SECUSER, AUTHSCIF, DSECUSER and DROPSCIF commands are closely related. The *userid* must have granted the issuing user authority through the AUTHSCIF command to use them as a secondary console. An error message will result if this has not been done. For example, if the *userid* WATCHER chooses to become the secondary console of the disconnected user SERVER, this sequence of commands is needed:
  - a. WATCHER issues DIRM AUTHSCIF SERVER
  - b. SERVER issues DIRM SECUSER WATCHER

When the secondary user facility is no longer needed, one of two commands would be needed:

- a. WATCHER issues DIRM DSECUSER SERVER
- b. SERVER issues DIRM SECUSER OFF

If SERVER is not going to use WATCHER as a SCIF user again, the general authority can be cancelled if WATCHER issues:

```
DIRM DROPSCIF SERVER
```

The design allows the SCIF target *userid* (WATCHER) to have overall control of the usage of SCIF, in case of SERVER produced SCIF output which the target was not able to handle. SERVER has sole control of the time it is turned on, but WATCHER must authorize it first. If the SECUSER command is issued before the nominated *userid* has granted authority with the AUTHSCIF command, an error message will result.

## DSECUSER

►► DIRMaint ——— DSECuser — *userid* ►►  
                   └─ Prefix Keywords 1 ─┘

Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the DSECUSER operand of the DIRMAINT command to request that a user, who is currently using the command invoker as a secondary console, drop the entry from their CONSOLE statement.

### Operands

#### *userid*

the user who is currently pointing at the command issuer as a secondary console.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. The use of SECUSER, AUTHSCIF, DSECUSER and DROPSCIF commands are closely related. The *userid* must have granted the issuing user authority through the AUTHSCIF command to use them as a secondary console. An error message will result if this has not been done. For example, if the *userid* WATCHER chooses to become the secondary console of the disconnected user SERVER, this sequence of commands is needed:
  - a. WATCHER issues DIRM AUTHSCIF SERVER
  - b. SERVER issues DIRM SECUSER WATCHER

When the secondary user facility is no longer needed, one of two commands would be needed:

- a. WATCHER issues DIRM DSECUSER SERVER
- b. SERVER issues DIRM SECUSER OFF

If SERVER is not going to use WATCHER as a SCIF user again, the general authority can be cancelled if WATCHER issues:

```
DIRM DROPSCIF SERVER
```

The design allows the SCIF target *userid* (WATCHER) to have overall control of the usage of SCIF, in case of SERVER produced SCIF output which the target was not able to handle. SERVER has sole control of the time it is turned on, but WATCHER must authorize it first. If the SECUSER command is issued before the nominated *userid* has granted authority with the AUTHSCIF command, an error message will result.

## DUMP



### Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

## Purpose

Use the DUMP operand of the DIRMAINT command to request a variable dump of the DIRMAINT virtual machine.

## Usage Notes

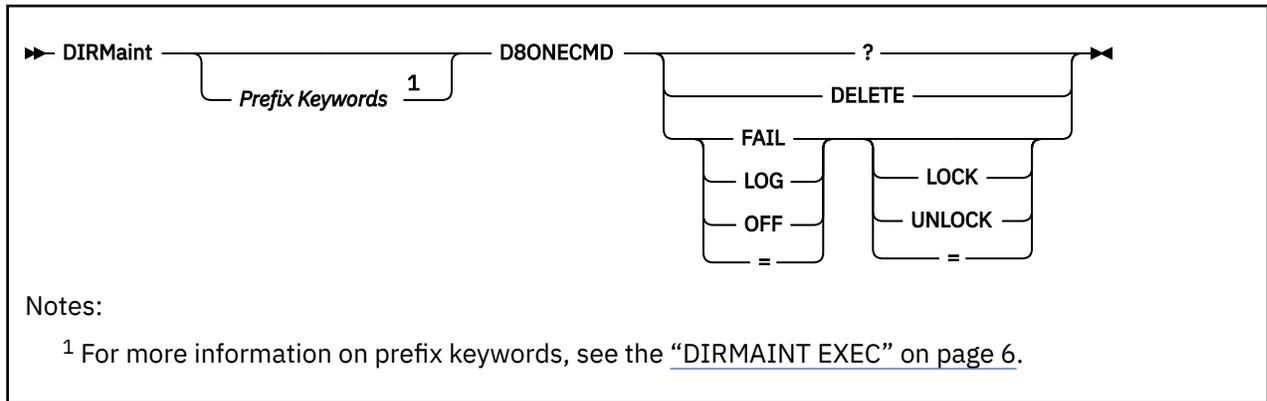
1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. A storage dump is not taken for Release 5. Instead several values are queried and returned to the issuer of the command as a file. The queries that are issued are:

```

QUERY TIME
IDENTIFY
QUERY FILES *
QUERY USER
QUERY SECUSER
QUERY CPLEVEL
QUERY CPU ID
QUERY V STORAGE
QUERY SET
QUERY CON
QUERY UR
QUERY PF
QUERY CMSREL
QUERY CMSLEVEL
QUERY EXECMAP
QUERY NUCXMAP
TYPE DVHSLVL DATADVH *
TYPE DVHVLVL DATADVH *
GLOBALV LIST
DVHSERVL

```

## D8ONECMD



### Purpose

Use the D8ONECMD operand of the DIRMAINT command to allow the user to query, alter, or delete the current D8ONECMD directory statement in a profile, user, or identity entry.

### Operands

?

allows you to query your current D8ONECMD settings.

#### DELETE

allows you to delete the existing D8ONECMD statement.

#### FAIL

tells CP to prevent this virtual machine from issuing multiple commands with DIAGNOSE X'08', the first command is processed, subsequent commands are rejected.

#### LOG

tells CP that this virtual machine can issue multiple commands through DIAGNOSE X'08', logging is performed.

#### OFF

tells CP that this virtual machine can issue multiple commands through DIAGNOSE X'08', logging is not performed.

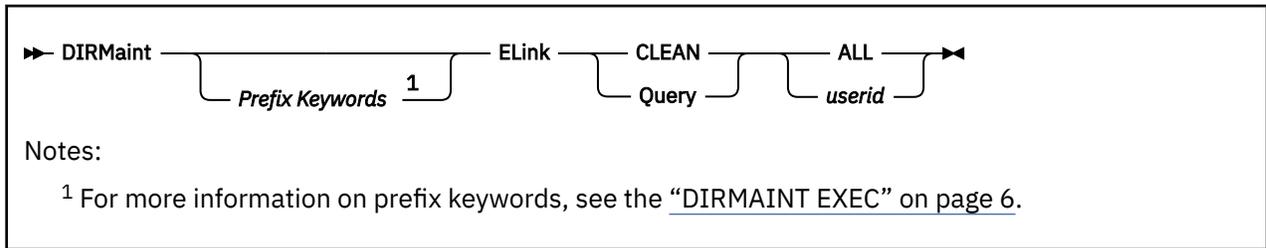
#### LOCK

#### UNLOCK

tells CP to lock or unlock the D8ONECMD setting for a specific entry.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the D8ONECMD directory statement see [z/VM: CP Planning and Administration](#).



## Purpose

Use the ELINK operand of the DIRMAINT command to authorize users to:

- Restore a user's authorization to issue the LINK operand
- Perform maintenance on the LINK ATTEMPTS file
- Report status of all or any entries in the LINK ATTEMPTS file.

## Operands

### CLEAN *userid*

specifies the user ID whose DIRM LINK privileges are being restored. This is accomplished by removal of the user's record in the DVHLINK ATTEMPTS file.

### CLEAN ALL

specifies that each record in the DVHLINK ATTEMPTS file is inspected. A record is only removed when the corresponding user has not issued an invalid DIRM LINK command in *nnn* days. The time period, *nnn* days is specified as the reset value in the LNKLMT statement in the CONFIG\* DATADVH file(s).

### Query *userid*

gives the status of a specified user ID in the DVHLINK ATTEMPTS file.

### Query ALL

gives the status of all user IDs in the DVHLINK ATTEMPTS file.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The default value of LNKLMT= is 7 if not specified in the CONFIG\* DATADVH file.
3. The DIRM ELINK CLEAN ALL command is used internally. It is performed automatically as scheduled in the DIRMAINT DATADVH wakeup times file. This daily cleanup activity should be sufficient for maintenance of the DVHLINK ATTEMPTS file.
4. Global maintenance of the DVHLINK ATTEMPTS file can be accomplished by use of the DIRM SEND and DIRM FILE operands.

When global maintenance is performed, the privileged user should be aware of the format of the *date* field in the DVHLINK ATTEMPTS file. The value is not in *mm/dd/yy form*. It is expressed in the ISO standard date format *yyyymmdd*.

5. The

LNKLMT= *threshold reset\_days*

statement is a configuration statement that takes two parameters. The *threshold* defines the number of unsuccessful link attempts that are tolerated before locking. If missing or set to 0 this feature is

## ELINK

disabled. The *reset\_days* defines the number of days the user will be prohibited from issuing LINK commands. If missing, it defaults to 7 days.

## ENABLE



### Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the ENABLE operand of the DIRMAINT command to cancel a previous DISABLE operand, and allow incoming source update transactions to be accepted.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. This operand causes the DISABLE CONTROL file to be erased from the primary and secondary (if required) directory disks. The existence of this file disables updates to the directory.

## EXECDROP

---

►► DIRMaint — EXECDrop ◄◄

### Purpose

Use the EXECDROP command to undo the effects of an EXECLOAD command. It removes DirMaint processing routines from user resident storage. It also releases (and detaches) the minidisk or SFS directory on which the DirMaint interface files reside.

### Usage Notes

1. DirMaint prefix operands do not apply to this command. They are accepted without error, but otherwise ignored.
2. The EXECDROP command is completely handled within the virtual machine of the command issuer. No transaction is sent to the DIRMAINT service machine, therefore no password is required or requested for authentication, even if you have NEEDPASS set to YES.
3. The EXECDROP command is not valid within a BATCH command file.
4. A DIRM EXECLOAD should be preceded by a DIRM EXECDROP. This allows optimization of storage usage and ensures the interface disk or directory is re-accessed.
5. The files to be dropped are listed on the LOADABLE\_USER\_FILE= entries in the CONFIG\* DATADVH file(s).

## EXECLOAD

---

►► DIRMaint — EXECLoad ◄◄

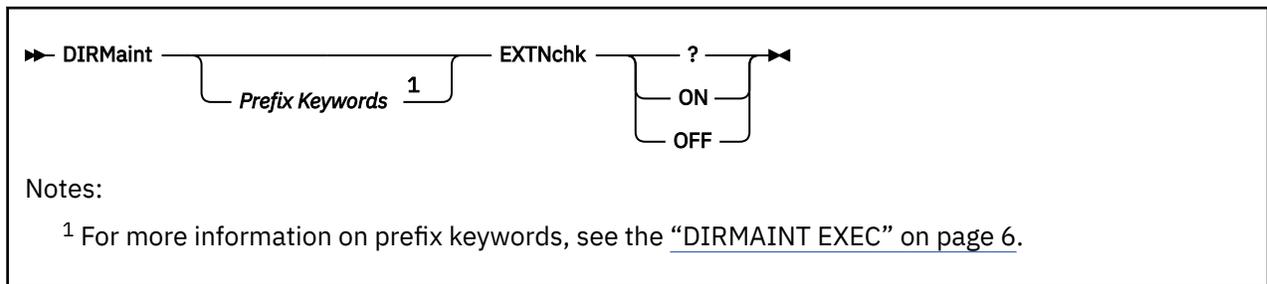
### Purpose

Use the EXECLOAD command to improve performance for those users who frequently issue DirMaint commands, such as the administration staff. It links and accesses the minidisk or SFS directory on which the DirMaint interface files reside, and loads the executable code (EXECs, XEDITs, MODULEs) into storage. The effects of EXECLOAD last until reversed by use of a DIRM EXECDROP command, you re-IPL CMS, or logoff.

### Usage Notes

1. DirMaint prefix operands do not apply to this command. They are accepted without error, but otherwise ignored.
2. The EXECLOAD command is completely handled within the virtual machine of the command issuer. No transaction is sent to the DIRMAINT service machine, therefore no password is required or requested for authentication, even if you have NEEDPASS set to YES.
3. The EXECLOAD command is not valid within a BATCH command file.
4. A DIRM EXECLOAD should be preceded by a DIRM EXECDROP. This allows optimization of storage usage and ensures the interface disk or directory is re-accessed.
5. The files to be loaded are listed on the LOADABLE\_USER\_FILE= entries in the CONFIG\* DATADVH file(s).

## EXTNCHK



### Purpose

Use the EXTNCHK operand of the DIRMAINT command to control the MDISK extent validation.



#### Attention:

Use extreme care when extent checking is not active. Data corruption can occur if MDISK extents are allowed to overlap.

### Operands

?

requests that the current extent check value be returned.

ON

specifies that when a new minidisk is being allocated manually and a specific extent is being requested, the DIRMAINT virtual machine verifies that the extent is really available.

**Note:** This is the default condition. It is not necessary to set this parameter ON unless your configuration file indicates OFF or you have previously issued this operand with the OFF value.

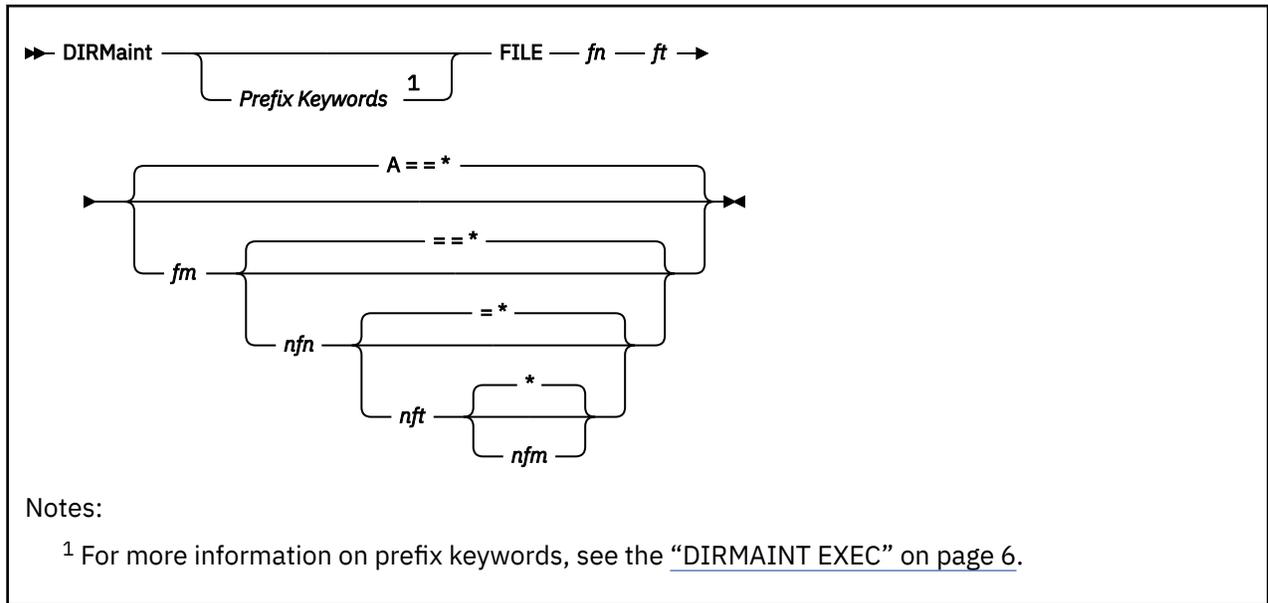
OFF

specifies that the validation of manually-allocated extents is inhibited. Use this operand if you deliberately want to allocate an overlapping minidisk.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The setting of EXTNCHK will be in effect for minidisks being added using explicit extent information.
3. The various automatic allocation methods provided by DirMaint can not be used to create overlapping minidisks. This setting is not consulted by these routines.

## FILE



## Purpose

Use the FILE operand of the DIRMAINT command to install or replace a file on one of DirMaint's minidisks.

## Operands

### *fn ft fm*

is the file identification and location on the invoker's minidisk. File mode defaults to A if no additional operands are needed.

### *nfn nft nfm*

is the file identification and location as it is to be stored inside DirMaint. The *nfn* and *nft* operands may be omitted (and default to =) if the invoker's copy of the file already has the correct file identification and *nfm* is not specified.

The *nfm* operand is usually omitted. The file will be stored automatically on the correct DirMaint minidisk according to file name and file type. If a file matching the new file name and new file type is not found on any disk in the DIRMAINT machine search order (excluding the backup disks), the new file will be placed on the:

- D disk for CONFIG\* DATADVH files
- E disk for \* PROTODIR files
- A disk for other files.

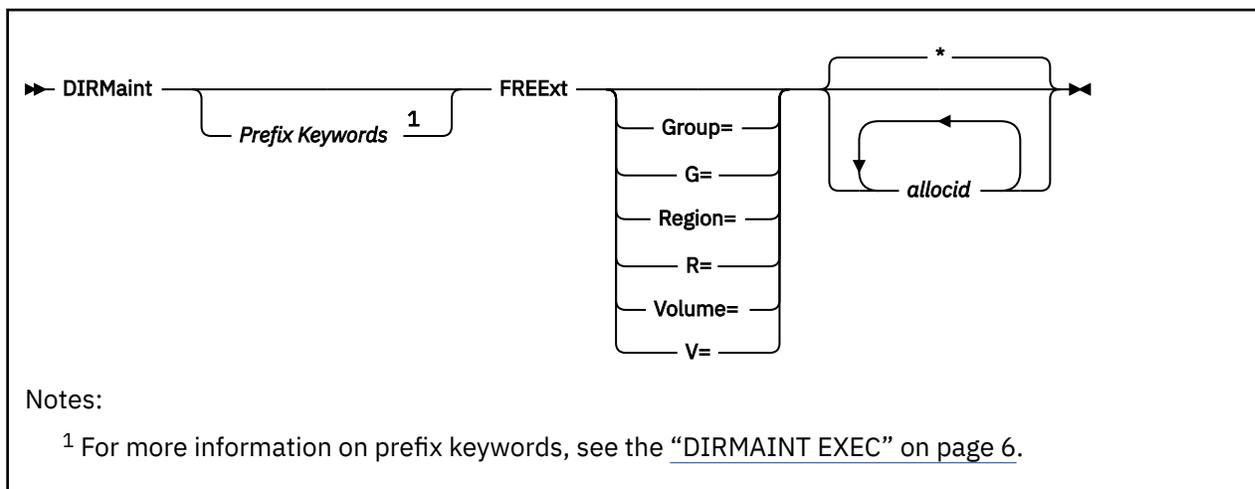
## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. If the FILE operand is used to replace a part listed on a LOADABLE\_SERV\_FILE= or LOADABLE\_DIRMAINT\_FILE= record in the CONFIG\* DATADVH file(s), then that operation should be followed with the RLDCODE operand to put the replaced file into operation. Otherwise, it is not placed in use until the next system restart.

## FILE

3. This operand cannot be used to maintain DIRMAINT EXEC, DIRMSAPI EXEC SAMP or ACCESS DATADVH. They must reside on the CMS system disk, its extension disk, or the licensed program disk.

## FREEXT



## Purpose

Use the FREEXT operand of the DIRMAINT command to generate a listing of free extents on volumes, groups, and regions used to contain minidisks.

## Operands

### *allocid*

specifies the search domain. If VOLUME= is specified, then *allocid* is the volume ID. If GROUP= is specified, then *allocid* is the group name. If REGION= is specified, then *allocid* is the region name.

If none are specified, then volume ID is assumed, but the reports will be returned in the format used before DirMaint function level 510. This older format is not intended to be used as a programming interface. See “Reports” on page 118 for the newer formats.

### volume ID

is the volume label of the volume(s) to be scanned, with each specific volume ID being 1 to 6 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +.

If a 1 to 6 character volume label is specified, the listing will contain only allocatable extents on that volume.

If a partially-qualified volume label of 1 to 5 characters followed by an asterisk (\*) is specified, the listing will include allocatable extents on all volumes beginning with the qualified volume ID. For example, if the total set of volumes containing minidisks are:

```
ABC
ABC12
ABC456
ABX445
VOL001
VSZ777
```

then specifying volume ID as ABC\* will cause volumes ABC, ABC12 and ABC456 to be scanned. Specifying ABC would qualify only the discrete volume ABC. Specifying V\* would cause volumes VOL001 and VSZ777 to be qualified.

If the volume ID is omitted, or if an \* is specified, then all volumes used to house minidisks will be included in the output.

Multiple discrete volumes (or multiple partially-qualified volumes) are also allowed.

**group name**

is the name specifying the group(s) to be scanned, with each specific group name being 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +. The same rules apply for partially-qualified group names as for volume IDs (above). Note that specifying group names on this command will result in a volume space report for each of those volumes containing regions that are part of those groups.

**region name**

is the name specifying the region(s) to be scanned, with each specific region name being 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +. The same rules apply for partially-qualified region names as for volume IDs (above). Note that specifying region names on this command will result in a volume space report for each of those volumes containing those regions.

**Reports**

Reports returned by the FREEExt command, with the G=, GROUP=, R=, REGION=, V=, or VOLUME= keyword specified, appear in the following format:

FREEEXT *operands*

GROUP	REGION	VOLUME	START	SIZE	(END)	OWNER	ADDR	SA
gggggggg	rrrrrrrr	ssssss	nnnnnnnnnn	nnnnnnnnnn	nnnnnnnnnn	.FREE.	0000	*
gggggggg	rrrrrrrr	ssssss	nnnnnnnnnn	nnnnnnnnnn	nnnnnnnnnn	.FREE.	0000	*

Unless the RETVIA/RETURNVIA prefix keyword is used, the result will be a reader file in the above format. There will be one header line per query request. There will usually be one detail line per free space entry. The "GROUP" column will be an "\*" for a REGION or VOLUME report. The "REGION" column will be an "\*" for a VOLUME report. The (END) column is calculated as (START+SIZE)-1.

Note that the output is similar to the USEDext report (see "USEDEXT" on page 281), except for the following:

- OWNER will be ".FREE."
- ADDR will be "0000"
- SYSAFFIN will be "\*"

If the free space is contained within multiple overlapping regions, or in a region that is included in more than one group, there may be multiple entries for that free space record.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, MENU, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The name of the file returned to the invoker of this command is dependant on the method used to invoke the command.

**allocid FREEEXT**

specifies the invocation was made with a specific group/region/volume name specified.

**alloc\$ FREEEXT**

specifies the invocation was made with a partially qualified group/region/volume specified.

**ALLGRPS FREEEXT**

specifies the invocation was made with a \* as the qualifier for group.

**ALLREGS FREEXT**

specifies the invocation was made with a \* as the qualifier for region.

**ALLVOLS FREEXT**

specifies the invocation was made with a \* as the qualifier for volume.

**MULTIGRP FREEXT**

specifies the invocation was made with more than one qualifier for group.

**MULTIREG FREEXT**

specifies the invocation was made with more than one qualifier for region.

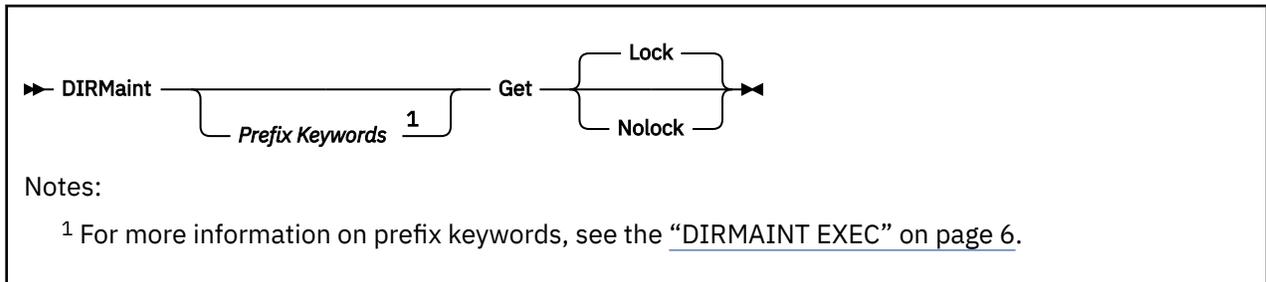
**MULTIVOL FREEXT**

specifies the invocation was made with more than one qualifier for volume.

**Note:** When there are no free extents or used extents meeting the search criteria the corresponding file returned will contain only a header with no data.

4. If the command invocation results in multiple groups/regions/volumes being queried (either by specific name specification or through the use of a partially qualified name specification) each name will appear in the report file sorted by that name.
5. If the volume architecture is CKD, free extents are considered to start at cylinder 1. If the volume architecture is FBA, free extents are considered to start at block 32.
6. In the normal case, the starting cylinder or block for one minidisk will be greater than the ending cylinder or block for the preceding minidisk on the volume. If the starting cylinder or block for any minidisk is less than or equal to the ending cylinder or block of the preceding minidisk, an exception is flagged. If the two minidisks have the:
  - Same starting and ending location, virtual disk address, and owner, differing only in the system affinity associated with the two entries (with neither system affinity being an asterisk), then the condition is flagged as shared.
  - Same starting and ending location, address, owner, differing only in the system affinity with one of them being an asterisk or if the system affinity values are also the same, then the condition is flagged as an error.
  - Different starting locations, or different ending locations, or different virtual addresses, or different owners, then the condition is flagged as an overlap. If any exceptions are flagged, a DIAGNOSE file will be generated in addition to the requested report file.
7. The "Special User IDs in the Default User Directory (USER DIRECT)" section in *z/VM: Installation Guide* recommends using and/or updating special dummy user IDs in the default user directory (USER DIRECT) to track allocation of CP areas on a volume. For example, the \$DIRECT\$ user ID tracks directory cylinders/blocks. The CP QUERY ALLOC MAP command identifies CP reserved areas, and these areas are tracked by DirMaint in the command output as .DRCT., .PAGE., .SPOOL., and .TDISK. areas. The FREEXT command does not list overlaps between a special user ID and the allocation extents information when the extents exactly match. This avoids an OVERLAP warning on those extents.

## GET



### Purpose

Use the GET operand of the DIRMAINT command to retrieve a copy of a profile, user, identity, or subconfiguration entry. With the entry on your own A-disk, you may introduce new control statements or modify existing ones using an editor. The updated directory entry must be restored to the directory using the REPLACE operand.

### Operands

#### Lock

is used to lock the directory entry after the GET processing has completed. This is the default.

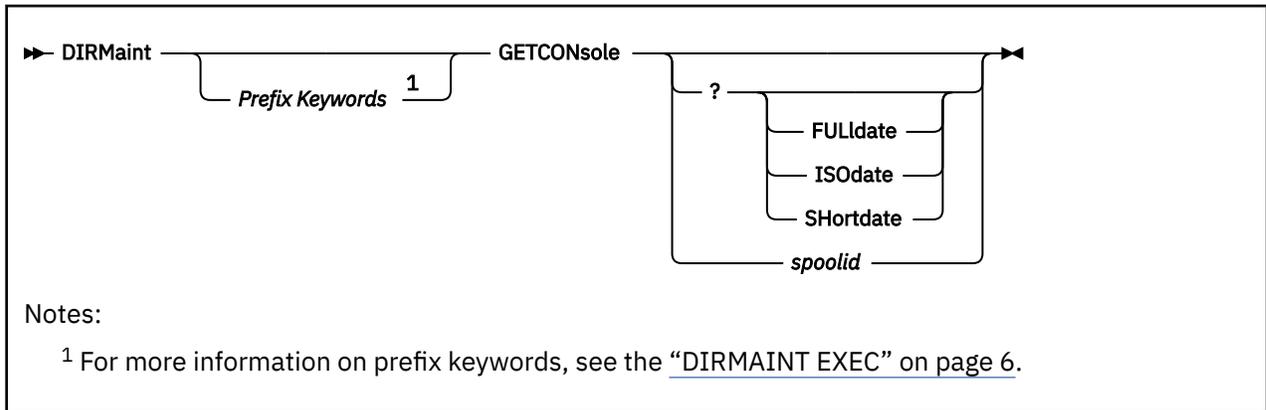
#### Nolock

is used if the request is for informational purposes only. No attempts are made to lock the directory entry.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, MENU, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The directory entry will be sent to your virtual card reader and you will be notified by a message. Use the CMS RECEIVE command to load it onto your minidisk. The file mode allocated defaults to A0.
4. If this request is for information only and you did not use the NOLOCK option, you should immediately issue the UNLOCK operand. On the assumption that the purpose of the GET was to modify some data item, the system has automatically locked the directory entry. This locking out is conveyed to you by a message. Other users are thus prevented from making any changes while you have a copy of the directory entry for modification.
5. The directory entry can now be reviewed in your own virtual machine and modified, as appropriate, with the CMS editor.  
The revised file can be put into effect by the DIRMAINT REPLACE operand.
6. The lock attempt (which is automatically generated by the GET request) on a directory entry with pending device locks will be rejected. This is done to prevent alterations to the entry while DASD operations are pending.
7. The default action of a GET request is to issue an implicit LOCK request. This action will succeed if the user issuing the GET is also authorized to LOCK user IDs. The LOCK transaction will fail, with appropriate error messages, if the user is not authorized for LOCK requests.

## GETCONSOLE



### Purpose

Use the GETCONSOLE operand of the DIRMAINT command to return a copy of the console logfile from the DIRMAINT machine.

### Operands

?

specifies that the list of available console files is to be displayed to the issuing user.

#### FULLdate

specify for the CP QUERT PRT command.

#### ISOdate

specify for the CP QUERT PRT command.

#### SHortdate

specify for the CP QUERT PRT command.

#### *spoolid*

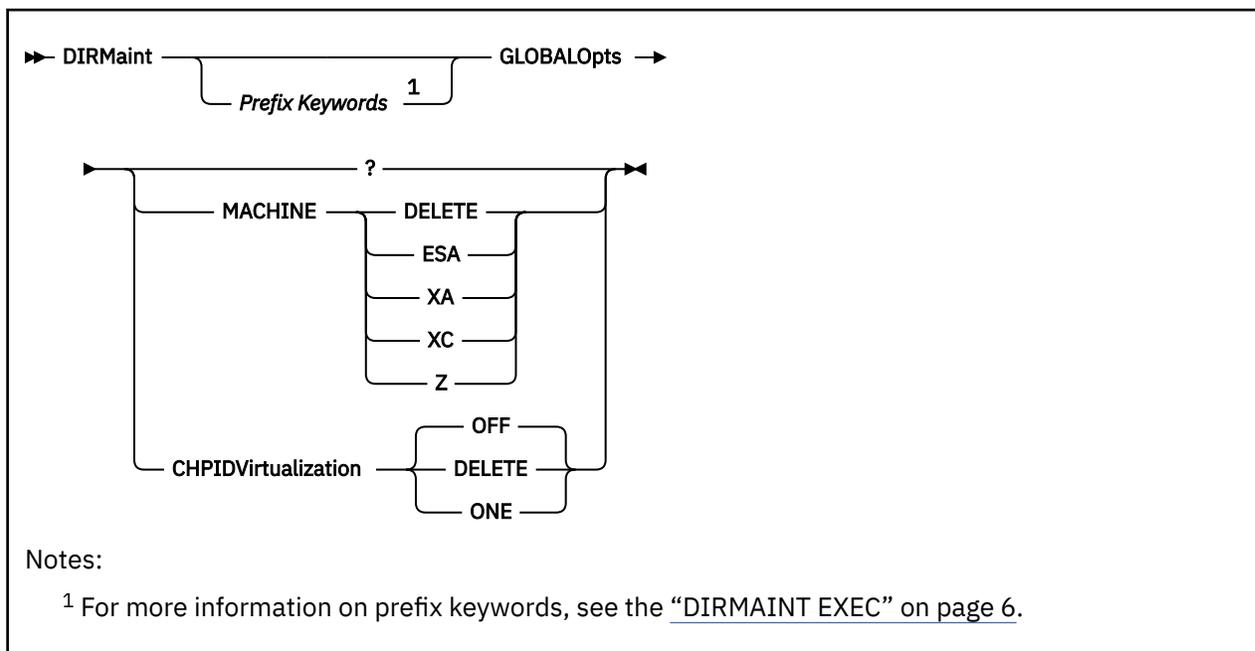
specifies the specific spool ID to be returned to the user. If *spoolid* is not specified, the currently active console file is closed and returned.

**Note:** Specifying the spool ID of the currently active console spool file will result in an error. To obtain a copy of the currently active console spool file, omit the *spoolid* parameter.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, MENU) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. For more information about the FULLDATE, ISODATE, or SHORTDATE operands, type HELP CP QUERY or see [z/VM: CP Commands and Utilities Reference](#).

## GLOBALOPTS



### Purpose

Use the GLOBALOPTS operand of the DIRMAINT to command add, delete, alter, or query the current setting of the GLOBALOPTS control statement.

### Operands

?

requests that the current settings of the GLOBALOPTS statement be returned in a message.

#### MACHINE

indicates that one of the following actions should occur on the MACHINE field of the GLOBALOPTS statement:

##### DELETE

specifies that the MACHINE field be deleted.

##### ESA

sets the MACHINE field to ESA.

##### XA

sets the MACHINE field to XA.

##### XC

sets the MACHINE field to XC.

##### Z

sets the MACHINE field to Z.

#### CHPIDVirtualization

specifies the CHPIDVirtualization setting for any user definition that does not contain an OPTION directory control statement with CHPIDVirtualization already specified, as follows:

##### DELETE

specifies that the CHPIDVirtualization field be deleted.

##### OFF

sets the CHPIDVirtualization field to OFF. This is the default.

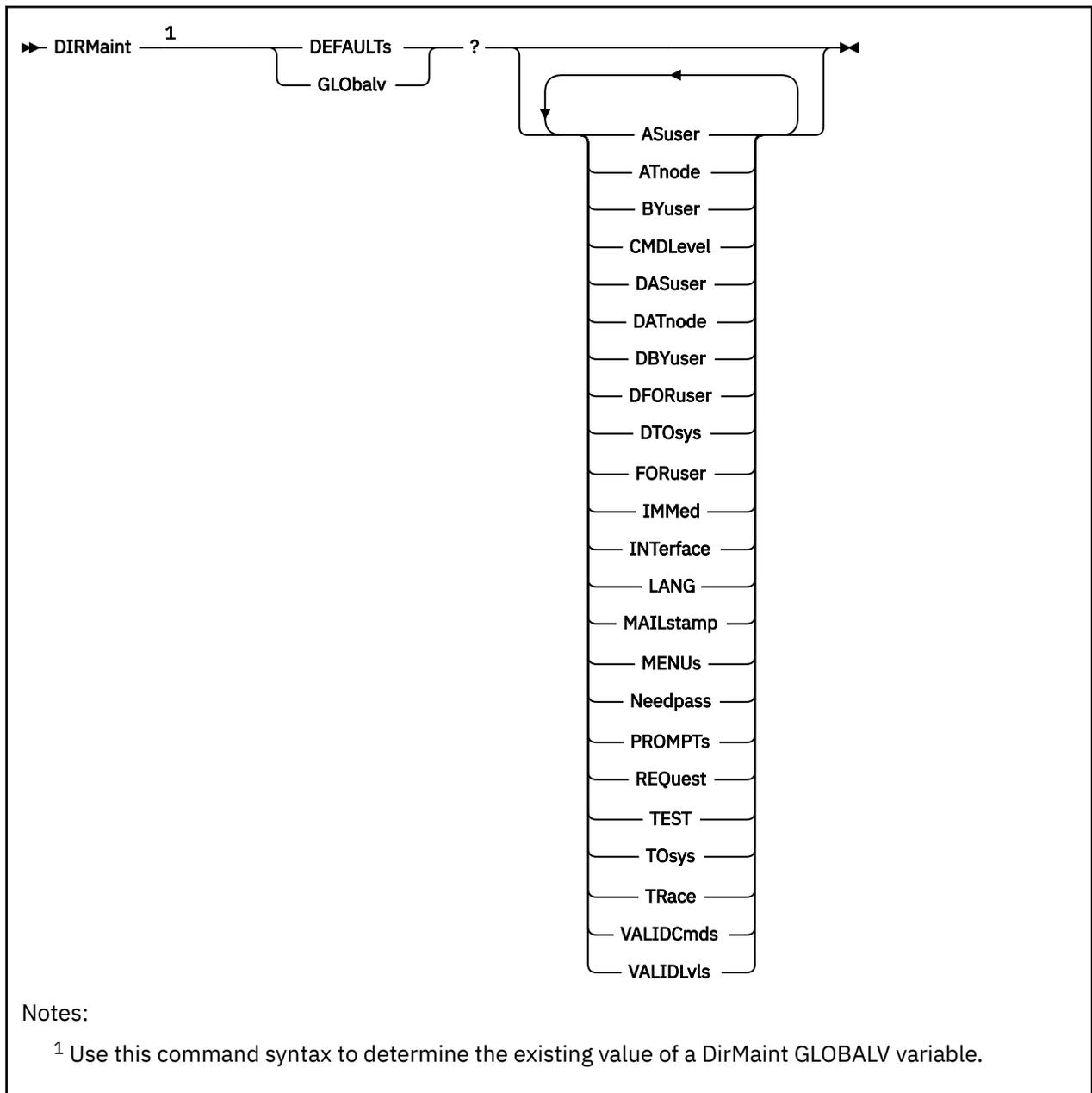
**ONE**

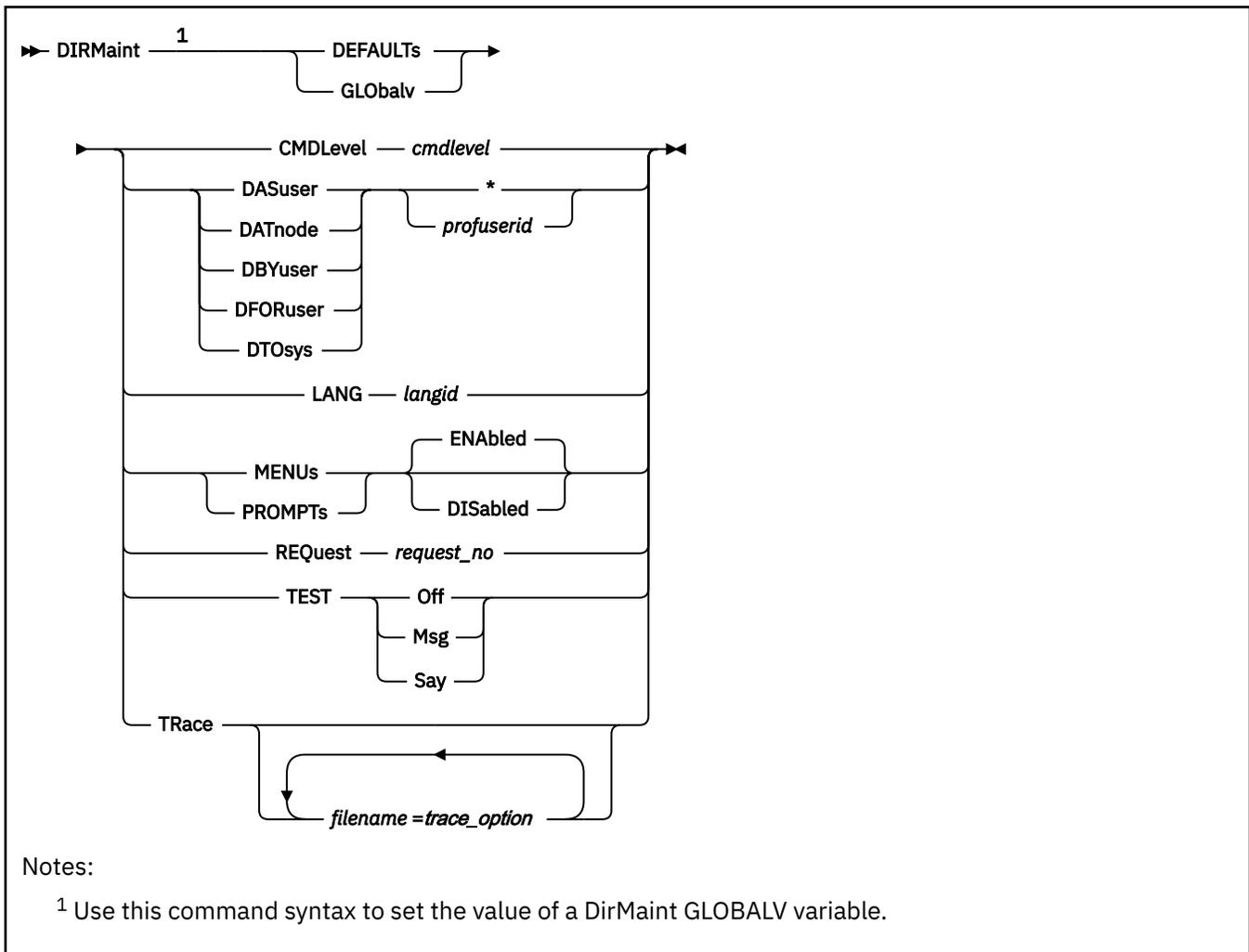
sets the CHPIDVirtualization field to ONE.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. These files will create the \$DIRGRP\$ DIRMPART file if none exists. Also, this file contains the GLOBALDEFS statement that is used to define the GLOBALOPTS and POSIXGROUP statements.
3. When a MACHINE statement is removed from a directory entry, the GLOBALOPTS machine definition will not become active until a DIRECTXA has been run against the full user directory file. To run DIRECTXA against the full user directory file, issue a DIRMaint DIRECT command. (See [“DIRECT”](#) on page 92.)
4. For more information on the GLOBALOPTS directory statement, see [z/VM: CP Planning and Administration](#).

## GLOBALV, DEFAULTS





## Purpose

Use the DEFAULTS or GLOBALV operand of the DIRMAINT command to set or query various default values used within DirMaint processing.

## Operands

?

Causes the current value of one or more variables to be displayed.

### ASuser

Is set by the program from the ASUSER prefix keyword on the DIRM command (if specified), or from the DASUSER global variable (if previously set), or to an \* by default.

### ATnode

Is set by the program from the ATNODE prefix keyword on the DIRM command (if specified), or from the DATNODE global variable (if previously set), or to an \* by default.

### BYuser

Is set by the program from the BYUSER prefix keyword on the DIRM command (if specified), or from the DBYUSER global variable (if previously set), or to an asterisk (\*) by default.

### CMDLevel *cmdlevel*

May be set by the user to indicate the compatibility level desired for use with application programs. The valid values are 150A or 140A.

**DASuser**

Identifies the default value for the ASUSER prefix keyword on the DIRMAINT command. The valid values are an asterisk or a valid file name which may be used as a user ID.

**DATnode**

Identifies the default value for the ATNODE prefix keyword on the DIRM command. The valid values are an asterisk or a valid file name which may be used as a network node ID.

**DBYuser**

Identifies the default value for the BYUSER prefix keyword on the DIRM command. The valid values are an asterisk or a valid file name which may be used as a user ID.

**DFORuser**

Identifies the default value for the FORUSER prefix keyword on the DIRM command. The valid values are an asterisk or a valid file name which may be used as a user ID.

**DTOsys**

Identifies the default value for the TOSYS prefix keyword on the DIRM command. The valid values are an asterisk or a valid file name which may be used as a network node ID.

**IMMed**

Is set by the program based on the presence or absence of the IMMED keyword prefix on the DIRM command.

**LANG *langid***

A valid *langid* consists of 5 characters or less, that are valid in a CMS file name, and is explicitly defined on one or more *langid* USER MSGS entries in the CONFIG\* DATADVH file(s).

Also a *langid* of *1SAPI* may be used to specify the *application programming interface* language, where all replies are:

```
REQUEST= nnnn RTN=DVHXXX MSGNO=1234 FMT=01 SUBS= VAR1 VAR2
```

**MAILstamp**

Displays the last time the mail file was sent to you.

**MENUs****ENABled**

specifies that a “fill in the blanks” menu will be displayed when required operands are not provided on a command invocation. This is the default.

**DISAbled**

prevents the display of the operand menu.

**Needpass**

The Needpass value is ignored and password prompting is not done for DirMaint commands when the NEEDPASS\_RESPECT configuration option is either allowed to default to NO or is specifically set to NO. When the NEEDPASS\_RESPECT configuration option is set to YES, the Needpass value indicates whether password prompting is done for most DirMaint commands (YES) or is not done (NO).

**PROMPTs****ENABled**

specifies that a prompt will be issued when a required password is not provided on a command invocation. This is the default.

**DISAbled**

prevents the prompt for the password.

**REQuest *request\_no***

Specifies an identifying number to be associated with a transaction sent to the DIRMAINT server. Used by programs using LANG 1SAPI to associate the transactions with the returned messages.

**TEST**

Determines whether or not a DIRM command results in a transaction to the DIRMAINT service machine (TEST OFF), or is displayed on the invoker's console as a CP message (TEST MSG), or is

displayed on the invoker's console using a CMS REXX SAY statement (TEST SAY). Valid values are OFF, MSG, or SAY.

**TOSys**

Is set by the program from the TOSYS prefix keyword on the DIRM command (if specified), or from the DTOSYS global variable (if previously set), or to an \* by default.

**TRace**

Is used for problem diagnosis to activate REXX tracing for the listed routines, and specify the degree of detail desired in the trace. Each entry must be in the form of a file name, followed by an equal sign, followed by a REXX trace option (All, Commands, Error, Failure, Intermediates, Labels, Normal, Off, Results, Scan); with no intervening blanks.

**VALIDCmnds**

Is set by the program to identify the valid values for CMDLEVELS. The valid values are 150A and 140A.

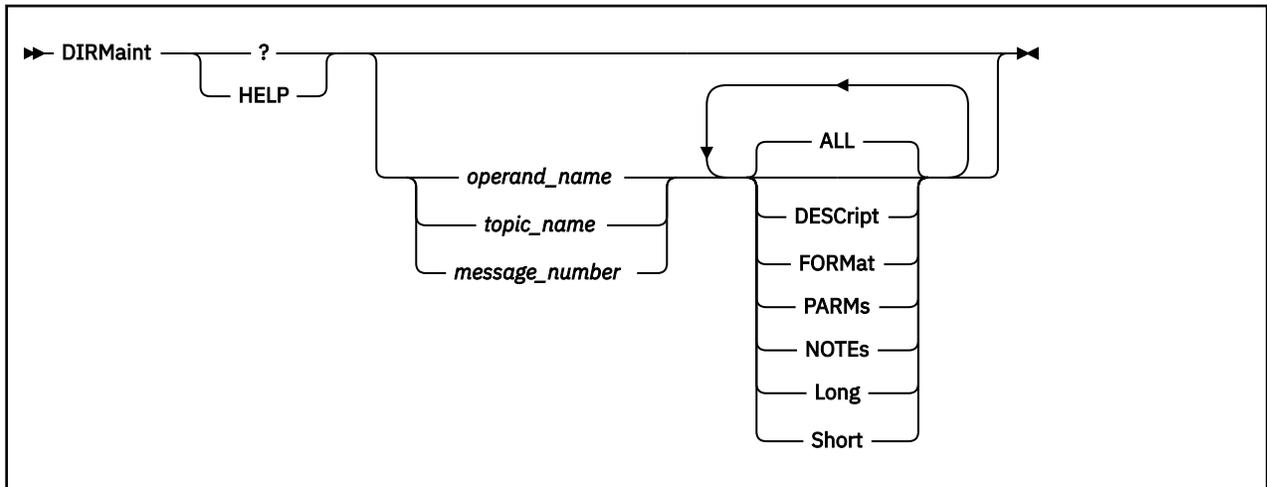
**VALIDLvls**

Is set by the program to identify the valid values for the interface level. The current value is 199501.

**Usage Notes**

1. DirMaint prefix operands do not apply to this command. They are accepted without error, but otherwise ignored.
2. The DEFAULTS and GLOBALV commands are completely handled within the virtual machine of the command issuer. No transaction is sent to the DIRMAINT service machine, therefore no password is required or requested for authentication, even if you have set NEEDPASS to YES.
3. The DEFAULTS and GLOBALV commands are not valid within a BATCH command file.
4. For more information on altering the NEEDPASS setting, see [“NEEDPASS” on page 171](#).
5. All global variables are stored in the DVH15 variable pool.

## HELP



## Purpose

Use the HELP operand of the DIRMAINT command to obtain online information about a DirMaint operand or message.

## Operands

### *operand\_name*

Is the name of any DirMaint operand on which information is desired. The available operands can be found by omitting the operand name and selecting one of the choices shown on a menu.

### *topic\_name*

Is the name of a DirMaint topic other than a DIRMAINT command on which help is desired. The available topics can be found by omitting the topic name and selecting one of the choices shown on the menu.

### *message\_number*

Is the identification of a DirMaint message.

The message number may be specified with or without the DVH prefix, routine ID, or the severity suffix. For example, DVH1119E, DVH1119, DVHADD1119E, DVHADD1119, 1119E, and 1119 are all valid.

### **ALL**

Displays all information available about the chosen operand, topic, or message.

### **DESCRiption**

Displays general description information only. For more information, see Usage Note “5” on page 129.

### **FORMat**

Displays format or syntax information only. For more information, see Usage Note “5” on page 129.

### **PARMs**

Displays parameter information only. For more information, see Usage Note “5” on page 129.

### **NOTEs**

Displays Usage Notes only. For more information, see Usage Note “5” on page 129.

### **Long**

Displays all HELP information available for the specified command, message, or topic. This is the same as specifying ALL.

**Short**

Displays an abbreviated version of the HELP file if it exists. Otherwise, the entire HELP files is displayed.

**Note:** DirMaint does not currently provide abbreviated HELP files.

**Usage Notes**

1. DirMaint prefix operands do not apply to this command. They are accepted without error, but otherwise ignored.
2. The HELP command is completely handled within the virtual machine of the command issuer. No transaction is sent to the DIRMAINT service machine, therefore no password is required or requested for authentication, even if you have used DIRM NEEDPASS to set NEEDPASS to YES.
3. The HELP command is not valid within a BATCH command file.
4. All DirMaint messages are prefaced with a message identification in the form of DVHrrrrnnns. For example, use DIRM ? DVH1232 for help on message DVHCMD1232E or DVHPAR1232E.

**Note:** Note: When using this form of the command, the prefix (DVH), routine ID (in this case, CMD or PAR) and any suffix (in this case, E) is optional.

Outside of DirMaint, you may also use CMS HELP xDVH DVH1234 for help on DVHCMD1234E or DVHPAR1234E, where x is A for AMENG, U for UCENG, or a unique letter to identify other languages.

5. When HELP is used for other than an operand name, the result is the same as using ALL.
6. Presuming that the DirMaint HELP files are installed on the same disk as the CMS and CP system HELP files, or that they are on an accessed minidisk or SFS directory, then the same result can be obtained by using the CMS system HELP command. Using the CMS HELP command requires specification of the language and component identification. Thus, DIRM ? MDISK becomes HELP ADVH MDISK for help in mixed case American English, or HELP UDVH MDISK for help in upper case English. The last three characters of the component code will be DVH for all DirMaint help files; the first character will usually be the initial letter of the name of the language. Check with your local support group to determine which languages are available, and the prefix letter for each.

**Examples**

1. Invoking all HELP information for the DIRMAINT APPCPASS operand.

```
DIRMAINT ? APPCPASS
```

2. Invoking only the syntax diagram for the DIRMAINT LINK operand:

```
DIRM HELP LINK FORMAT
```

3. Invoking all HELP information available for the DIRMAINT USEREXIT topic:

```
DIRM HELP USEREXIT
```

4. Invoking the HELP information for DIRMAINT message number DVH2119:

```
DIRM HELP DVH2119
```

## HISTORY

---

**Notes:**

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

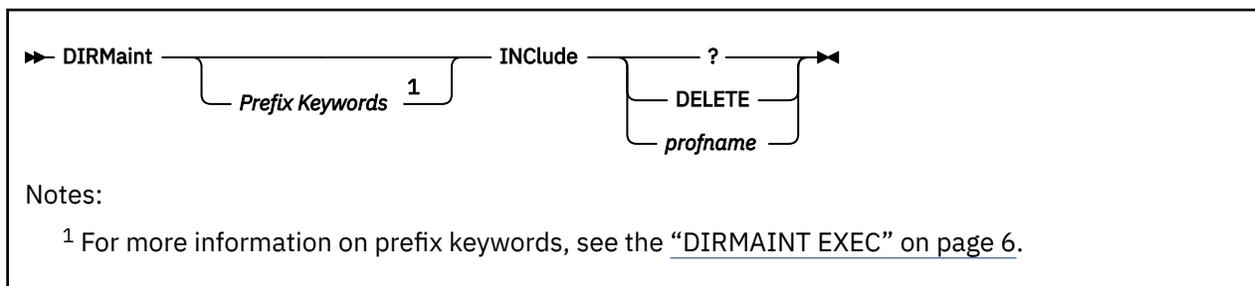
### Purpose

Use the HISTORY operand of the DIRMAINT command to request a copy of the history file.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The history file is a time-stamped sequential listing of all incoming operands.
3. The LOGMSG operand may be used to insert supplementary messages into the history file. For more information, see [“LOGMSG”](#) on page 150.
4. The disposition of older versions of the history file is determined in the DVHDAILY EXEC procedure.
5. The history file is identified on the MESSAGE\_LOGGING\_FILETYPE= configuration file entry.

## INCLUDE



### Purpose

Use the INCLUDE operand of the DIRMAINT command to allow the caller to query, alter, or delete the current INCLUDE directory statement.

### Operands

**?**

allows you to query the current INCLUDE statement.

#### **DELETE**

request to delete the current INCLUDE statement.

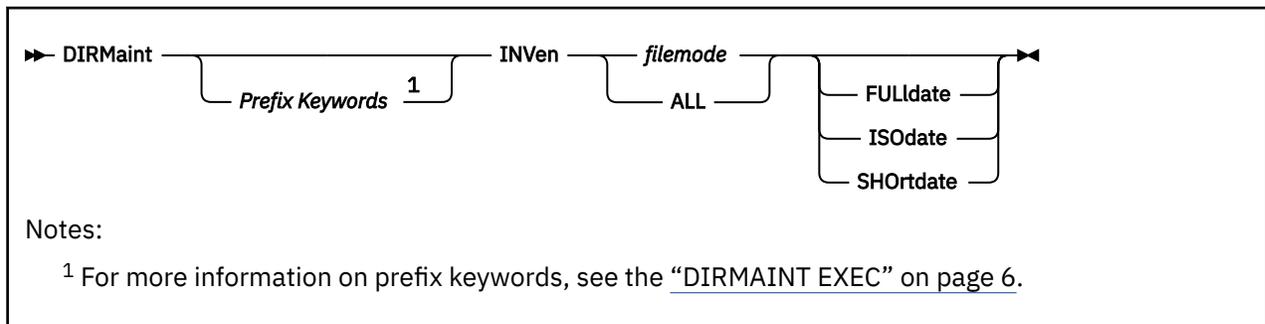
#### ***profname***

the name of the profile to be placed on the new INCLUDE statement.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the INCLUDE directory statement, see *z/VM: CP Planning and Administration*.
3. Because DELETE is a keyword, it cannot be used as a profile name. To use DELETE as a profile name, use the DIRM GET and DIRM REPLACE operands.

## INVEN



### Purpose

Use the INVEN operand of the DIRMAINT command to obtain an inventory of one or all of the DIRMAINT virtual machine's minidisks.

### Operands

#### *filemode*

specifies which of the minidisks you want to inventory.

#### ALL

Requests that all minidisks accessed R/W be inventoried.

#### FULLdate

specify for the LISTFILE command.

#### ISODate

specify for the LISTFILE command.

#### SHortdate

specify for the LISTFILE command.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The specified disk will be inventoried by executing the

```
CMS LISTFILE * * filemode ( DATE xxxDATE
```

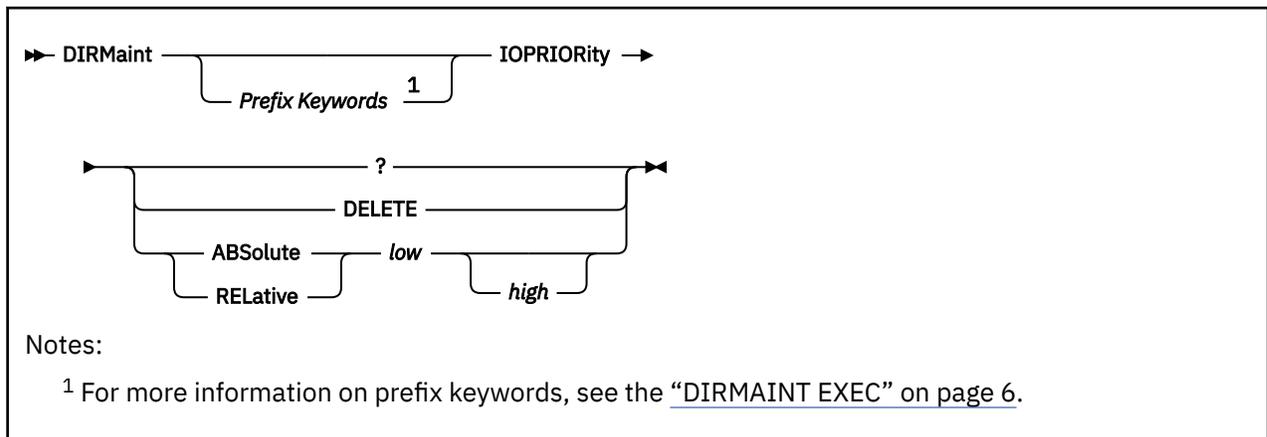
command. Where:

#### **xxxDATE**

This may be omitted or may be FULLDATE, ISODATE, SHORTDATE, or VMDATE if specified on the INVEN command.

3. The resultant file DIRMAINT DISKx will be sent to your virtual card reader. Use the CMS RECEIVE command to read the file.
4. If the disk is a CMS minidisk, a QUERY DISK is issued against the disk and appended to the file. If the disk is a SFS directory, a QUERY ACCESSED is issued against the file mode and appended to the file.
5. If the ALL operand is used, you will receive a separate file for each disk.
6. For more information about the FULLDATE, ISODATE, SHORTDATE, or VMDATE operands, type HELP CMS FILELIST or HELP CMS LISTFILE or see [z/VM: CMS Commands and Utilities Reference](#).

## IOPRIORITY



### Purpose

Use the IOPRIORITY operand of the DIRMAINT command to manipulate the CP IOPRIORITY directory statement in a profile, user, or identity entry. The IOPRIORITY statement defines a virtual machine's I/O priority queueing range.

### Operands

#### DELETE

specifies that the specified IOPRIORITY directory control statement should be deleted.

#### ABSolute

indicates the type of I/O priority range.

#### RELative

indicates the type of I/O priority range.

#### low

specifies the low value of the I/O priority range. It must be a number from 0 to 255

#### high

specifies 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.

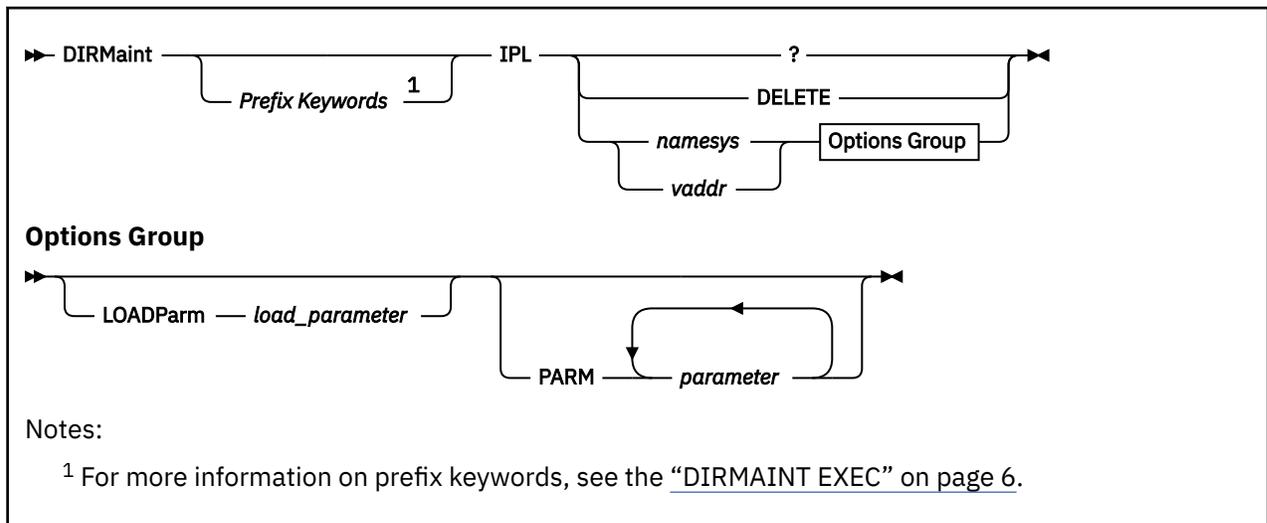
### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. If an IOPRIORITY statement is not found, LOGON processing requests RELATIVE I/O priority range with a low value of 0 and a high value of 0.
3. To assign a RELATIVE range of 8 values, use the following IOPRIORITY statement in a virtual machine's directory entry:

```
ioprior rel 0 7
```

4. For more information on the IOPRIORITY directory statement, see [z/VM: CP Planning and Administration](#).

## IPL



## Purpose

Use the IPL operand of the DIRMAINT command to add, delete, query, or change your directory entry IPL control statement. This statement specifies that CP automatically load your virtual machine when you log on to the system. The PARM option may be included for passing parameters to the IPL'd system.

Either `namesys` or `vaddr` will add an IPL control statement to your directory entry or will replace a statement that currently exists.

## Operands

**?**

requests that the current setting of your IPL statement be queried.

### DELETE

requests that the current IPL statement is to be deleted.

### namesys

is the name of a saved system to be automatically loaded each time you log on.

### vaddr

is the virtual device address of the device containing the system to be loaded.

### LOADParm *load\_parameter*

specifies a 1 to 8 character load parameter that is used by the IPL'd system.

### PARM *parameters*

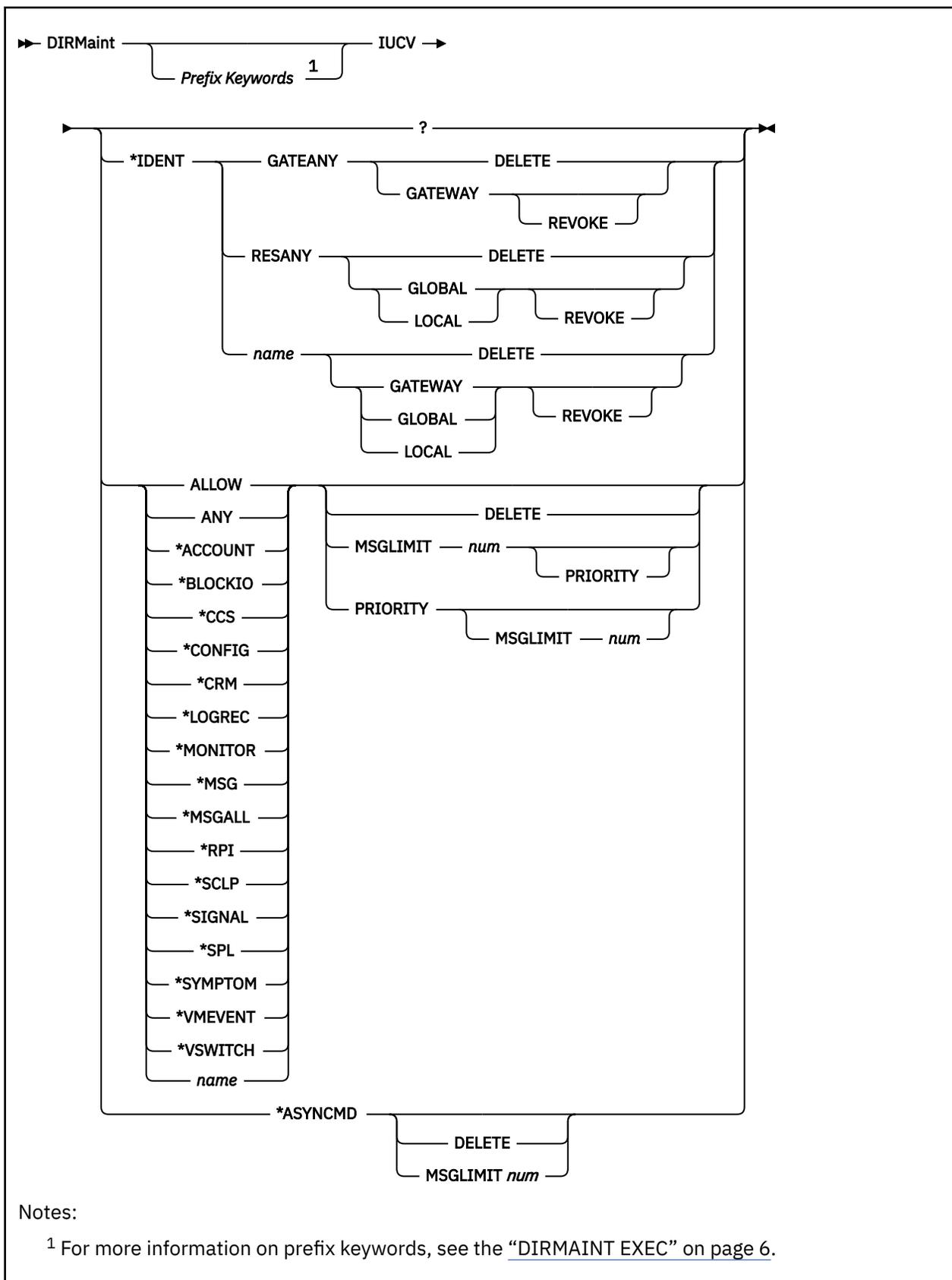
specifies the parameters to be passed to the IPL'd operating system, separated by blanks. Refer to Usage Notes [“4”](#) on page 134, [“5”](#) on page 135, and [“8”](#) on page 135.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the IPL directory statement, see *z/VM: CP Planning and Administration*.
3. The directory IPL control statement can be overridden at logon time by specifying LOGON *userid* NOIPL.
4. If multiple blanks are inserted between parameters, they are compressed to one blank.

5. The entire PARM string is placed into the directory provided it fits in 72 columns. Characters after column 72 are truncated.
6. Since the parameters specified on the PARM parameter are only meaningful to the system to be loaded, DirMaint does not validate them. The data is passed to the operating system that is receiving control as a result of the IPL process.
7. Because DELETE is a DirMaint keyword, it cannot be used as the name of a saved system on the IPL command. To use DELETE as a saved system name, use GET and REPLACE processing.
8. Because DIAGNOSE X'84' processing handles only the first 48 bytes of information, the PARM string is left justified and truncated to 48 bytes before being passed to the DIAGNOSE X'84' routine.
9. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

# IUCV



Notes:

<sup>1</sup> For more information on prefix keywords, see the “DIRMAINT EXEC” on page 6.

## Purpose

Use the IUCV operand of the DIRMAINT command to add, delete, change, or query IUCV directory statements in a profile, user, or identity entry.

## Operands

**?**

requests that the current settings of the IUCV statements be queried.

**\*IDENT**

allows this virtual machine to connect to the Identify System Service to identify a resource.

**GATEANY**

allows the virtual machine to identify any gateway LU name.

**RESANY**

allows the virtual machine to identify any resource name.

*name*

a 1 to 8 character name representing the gateway name, resource ID, gateway LU or user ID associated with the IUCV statement.

**DELETE**

requests that the specified IUCV statement be deleted.

**GATEWAY**

authorizes the virtual machine to identify the gateway LU.

**LOCAL**

authorizes the virtual machine to identify the resource as a local resource.

**GLOBAL**

authorizes the virtual machine to identify the resource as a global resource.

**REVOKE**

allows the virtual machine to revoke the specified resource or gateway LU name without owning it.

**ALLOW**

specifies that any other virtual machine can establish a communication path with this virtual machine.

**ANY**

is a general authorization indicating that a communications path can be established with any other virtual machine, resource, or gateway on this system.

**\*ACCOUNT**

Accounting system service.

**\*BLOCKIO**

DASD block I/O system service.

**\*CCS**

SNA Console Communications Service system service.

**\*CONFIG**

Configuration system service.

**\*CRM**

Collection resource management system service.

**\*LOGREC**

Error recording system service.

**\*MONITOR**

\*MONITOR system service.

**\*MSG**

Message system service.

**\*MSGALL**

Message all system service.

## IUCV

### **\*RPI**

Access verification system service.

### **\*SCLP**

SCLP system service.

### **\*SIGNAL**

Signal system service.

### **\*SPL**

Spool system service.

### **\*SYMPTOM**

Symptom system service.

### **\*VMEVENT**

Virtual Machine Event system service.

### **\*VSWITCH**

Switch system service.

### **MSGLIMIT *num***

defines the maximum number of outstanding messages allowed on any path authorized by this entry.

### **PRIORITY**

indicates that a communication path with the specified virtual machine can handle priority IUCV interrupts.

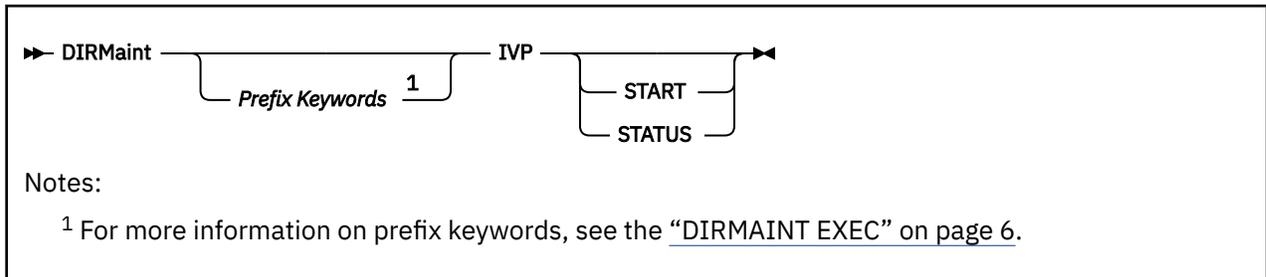
### **\*ASYNCMD**

Asynchronous command system service.

## **Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Not all options of this command are appropriate for z/VM. For more information on the IUCV directory statement, see *z/VM: CP Planning and Administration*.
3. As the IUCV statement can be used to control communication between virtual machines, careful consideration must be given to the security implications of using this statement.

## IVP



### Purpose

Use the IVP operand of the DIRMAINT command to start an automated run of the installation verification procedure (IVP) or to obtain status information on the run.

### Operands

#### IVP

starts an automated installation verification program, or obtains status information on the run.

#### START

starts an IVP run. A spool file containing the results of the run is sent to the user that issues the IVP START command. See [“IVP Results”](#) on page 139.

#### STATUS

requests status of the most recent IVP run. A spool file containing the status of the run is sent to the user ID that issues the IVP STATUS command. See [“IVP Results”](#) on page 139.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. The IVP is run against the DIRMAINT server and active SATELLITE and DATAMOVE servers. The IVP does not complete until the DIRMAINT server has received responses from the other servers, or until the run has reached the maximum configured time limit. For more information, see the description of the IVP\_MAX\_DURATION configuration property under "Step 6. Select IVP Characteristics" in the "Tailoring the DIRMAINT Service Machine" chapter in *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*.
3. See [“DATAMOVE”](#) on page 81 and [“SATELLITE ”](#) on page 224 for information on running the IVP only on those servers.
4. The IVP does not run on servers that are logged off; this information is noted in the IVP status file.
5. The IVP can be run from the console of the DirMaint server. The results are sent to the console instead of being sent to the user who initiated the run.
6. By default, the IVP is automatically run as part of new day processing on the DirMaint server.
7. You can override the default settings associated with the IVP by updating configuration statements in the CONFIG DATADVH file. For more information, see "Step 6. Select IVP Characteristics" in the "Tailoring the DIRMAINT Service Machine" chapter in *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*.

### IVP Results

**Note:** An IVP status file is created when an IVP run completes, or when the IVP STATUS command is specified. When an IVP run completes, a file called IVP RESULTS is sent to the user who issued the IVP

command. When an IVP STATUS command is issued, a file called IVP STATUS is sent to the user who issued the IVP STATUS command. The files have the same format; the files have different file types so that you can identify how they were created. (In this section, the terms "results" and "status" are used interchangeably.)

The following examples represent an IVP status file.

The first section of the status file, shown in [Figure 1 on page 140](#), contains information on the overall IVP run (for all servers). The line numbers are included for readability; they do not appear in the actual file.

```

1 Overall status: COMPLETED
2
3 Started: 10:43:21.915785 2020/09/08
4 Ended: 10:43:29.788387 2020/09/08
5
6 DirMaint Server:
7 DIRMAINT/TESTA, IVP Status: COMPLETED, 0 failures, 1 warnings
8
9 Number of satellite servers: 2
10 DIRMSATB/TESTB, IVP Id: 2, IVP Status: COMPLETED, 1 failures, 2 warnings
11 DIRMSATA/TESTA, IVP Id: 3, IVP Status: DISABLED
12
13 Number of datamove servers: 2
14 DATAMOVE/TESTA, IVP Id: 4, IVP Status: COMPLETED, 1 failures, 1 warnings
15 DATAMOV2/TESTB, IVP Id: 5, IVP Status: COMPLETED, 1 failures, 1 warnings

```

*Figure 1. Example of IVP Status File – First Section*

The number of lines in the first section will vary depending on the number of machines and status of the run. The lines in this sample contain the following information:

- 1** Overall status of the IVP run. This value can be STARTED or COMPLETED.
- 3** Start time of the IVP run.
- 4** End time of the IVP run. This line will not be displayed if the run is not complete.
- 6-7** Information about the main DIRMAINT server. This includes the following:
  - The user ID and node ID of the server.
  - The status of the IVP run on the server, (STARTED or COMPLETED).
  - The number of failed test cases.
  - The number of test cases that generated a warning. This number is not necessarily the same as the number of failed test cases. Some tests generate a warning but the problem is not severe enough to cause the test case to fail. An example of this is a test to warn of bad practices that will be deprecated and cause a failure in the future.
- 9** The number of SATELLITE servers.
- 10-11** A status line for each SATELLITE machine. Each line is in the same format as that for the DIRMAINT server (Line 7). Servers that have started an IVP run will have a status of STARTED or COMPLETED, and will include information on the number of tests that have failed and not failed (that either passed or are still running) which have generated at least one WARNING. The other status values are:
  - IVP\_DISABLED**  
Indicates testing of the server was disabled by the IVP\_NO\_TESTING configuration statement, or indicates this is the SATELLITE server that is intended to run on the same node as the DIRMAINT server and only when it is logged off.
  - LOGGED\_OFF**  
Indicates the server was logged off at the start of the IVP test run. The IVP will not log on the server because the installation may have logged it off for a reason.

**NOT\_LOADED**

Indicates that the server appears in a CONFIG\* DATADVH file but is not in the active control file. This situation can occur if you updated the configuration file but forgot to issue RLDDATA to reload the settings from the file. Restarting the DIRMAINT server reloads the settings.

**OVERDUE**

Indicates that an IVP request was sent to the server but the server did not respond with its completion information prior to reaching the maximum time duration for an IVP run.

**STARTED**

Indicates that an IVP request was sent to the server but the server has not yet responded, and the maximum time duration for an IVP run has not been reached.

**13**

Number of DATAMOVE servers.

**14-15**

A status line for each DATAMOVE machine. Each line is in the same format as that for the DIRMAINT server (Line 7).

The next section of the status file contains information on the DIRMAINT server, followed by sections for each of the other servers. The first part of this section, as shown in [Figure 2 on page 141](#), contains summary information related to the run. The following example shows information for the DIRMAINT server.

```

1 =====
2 Dirmaint: DIRMAINT/TESTA
3
4 IVP status: COMPLETED
5 Started: 10:43:21.915785 2020/09/08
6 Ended: 10:43:22.105419 2020/09/08
7
8 Number of tests planned: 19
9 Number of tests attempted: 19
10 Number of tests completed: 19
11 Number of failed tests: 0
12 Number of non-failed tests with warnings: 1
13
14 The following tests were disabled:
15     DM1000
16
17 The following tests failed:
18     DS0100
19
20 The following tests did not fail but generated warnings:
21     DM0110

```

*Figure 2. Example of IVP Status File – DIRMAINT Server Section*

The number of lines in the above section will vary depending on any tests were disabled or generated warnings. The lines in this sample contain the following information:

**1**

Section separator.

**2**

Type of server (DIRMAINT, SATELLITE, or DATAMOVE) and server identifier information (user ID/node).

**4**

Status of the IVP run on the server. This can be COMPLETED, LOGGED\_OFF, NOT\_LOADED, OVERDUE, or STARTED.

**5**

Start time of the IVP run on the server.

**6**

End time of the run on the server. This line will not be displayed if the run is not complete.

**8**

The number of possible tests for this type of server.

**9**

The number of tests attempted. This value can be less than the number of possible tests because tests were disabled or not run.

**10**

The number of tests that completed.

**11**

The number of tests that encountered a significant error that caused them to fail. Some warnings are minor and do not cause a test to fail.

**12**

The number of tests that passed but generated warnings.

**14-15**

the list of test cases that were disabled by a configuration file.

**17-18**

the list of test cases that failed. These lines are not displayed if no tests failed.

**20-21**

The list of test cases that passed but generated a warning. These lines are not displayed if no tests passed and generated a warning.

The next section of the status file, as shown in [Figure 3 on page 142](#), provides information on individual tests as each test is attempted (or skipped because it has been disabled). Each test that is started contains at least two lines; additional lines might be added as part of the test to aid in problem analysis. When the test completes a status line is added. Note that the test identifiers (IDs) shown (such as DM0020) are a unique identification of a test procedure that identify the procedure and where it is being run. For example, DM0020 is run on the DirMaint server. There are other versions of the test that run on DATAMOVE and SATELLITE servers; they are called DA0020 and DS0020, respectively. The order that test IDs appear in the status file is not significant and could change.

```

1 DM0020: Starting validation
2 Description: Identify the accessed disks.
3 The following disks were accessed.
4   FM PURPOSE  ADDR USED OWNERID ADDR / or SFS Directory
5   -----
6   A   RWS      155  45  DIRMAINT/0155
7   C   SRV      191  32  DIRMAINT/0191
8   D   USR      11F  24  DIRMAINT/011F
9   E   PDF      1DF   7  DIRMAINT/01DF
10  F   SSI      551  18  PMAINT/0551
11  G   PDB      1DB   5  DIRMAINT/01DB
12  S                190  43  MAINT/0190
13  Y/S            19E  31  MAINT/019E
14  Z   SFA      1FA   2  DIRMAINT/01FA
15  Review the DVHPROFA DIRMAINT file on the server for more information
17  on the disks and their purpose.
18 Status: PASSED

```

*Figure 3. Example of IVP Status File – Individual Test Section for a Test that Passed*

The number of lines in the above section will vary depending on any tests were disabled or generated warnings. The lines in this sample contain the following information:

**1**

Indicates the test ID and shows that this test was started.

**2**

Brief description of the test.

**3-17**

Additional information identified by the test to aid in analysis. (Most tests that pass do not generate additional information.)

**18**

Status of the test. This can be FAILED, PASSED, SKIPPED, or STARTED.

Figure 4 on page 143 is an example of a test that passed but for which non-failure warnings were generated. In this example, PROTODIR files were expected to be on the PDF (E) disk but were found on other disks.

```

1 DM0110: Starting validation
2 Description: Verify * PROTODIR files are only on the PDF (E) disk.
3 WARNING: The following files were found on unexpected disks.
4   Filename Filetype  FM  DATE      TIME
5   -----
6   P1          PROTODIR A0  9/08/20  10:43:23
7 Status: PASSED

```

Figure 4. Example of IVP Status File – Individual Test Section for a Test that Passed with Warnings

Note that the lines of output are similar to those described in Figure 3 on page 142.

Figure 5 on page 143 is an example of a test that was skipped because it was disabled by the IVP\_DISABLE\_TEST configuration property.

```

1 DM0430: Starting validation
2 Description: Begin IVP on satellite servers.
3 Status: SKIPPED

```

Figure 5. Example of IVP Status File – Individual Test Section for a Test that was Skipped

Note that the lines of output are similar to those described in Figure 3 on page 142.

Figure 6 on page 143 is an example of a test to run the IVP on two DATAMOVE servers. The IVP for the two servers will complete asynchronously from each other.

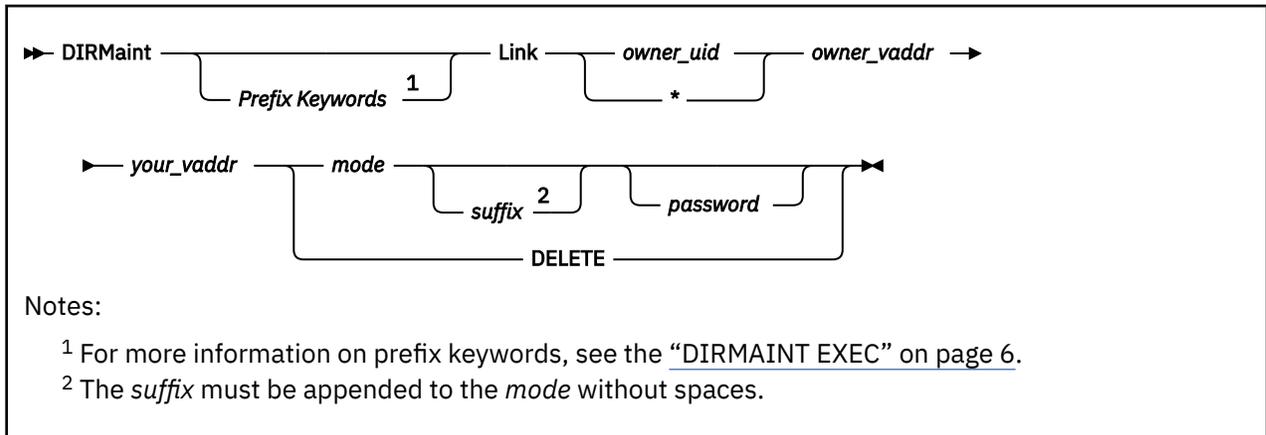
```

1 DM0440: Starting validation
2 Description: Begin IVP on datamove servers.
3   Starting IVP on server: DM012/DVHTST1
3   Starting IVP on server: DM011/DVHTST1
4   IVP validation on the servers will complete asynchronously.
5   DM0440 will appear again in the status output with updated
6   completion status when all datamove servers have completed their
7   DM0440 will appear again in the status output with updated
8   IVP runs.
3 Status: STARTED

```

Figure 6. Example of IVP Status File – Individual Test Section for a Test that was Started

## LINK



## Purpose

Use the LINK operand of the DIRMAINT command to allow you to request that a LINK control statement be inserted in your directory entry and put into effect automatically each time you log on. This makes it possible for your virtual machine to automatically establish a link to another virtual machine's minidisk without you needing to issue the CP LINK command during your terminal session.

## Operands

### *owner\_uid*

specifies the user ID of the minidisk owner to which the requestor wishes to link.

\*

indicates the user ID is added to the LINK control statement in your user entry as the target minidisk owner. For more information, see the description of the LINK directory statement in [z/VM: CP Planning and Administration](#).

### *owner\_vaddr*

specifies the virtual address of the minidisk as addressed by the owner.

### *your\_vaddr*

specifies the virtual device address you will use to access the minidisk.

### *mode*

specifies the type of access you have to the minidisk when you log on. The owner must have assigned a password corresponding to the mode you are requesting or must have granted specific permission to link to the device using the AUTHLINK command. For example, if the owner has established a read password or authorized a read link, then you can request a mode of R or RR. Valid modes are:

#### Mode

##### Definition

#### R

Read-only (R/O) access is desired. Access is not allowed if the owner or any other user has a link to the minidisk in write or any exclusive status.

#### RR

Read-only (R/O) access is desired, even if the owner or another user has a link to the minidisk in write status. Access is denied if any user has the minidisk linked in exclusive status.

#### W

Write access is desired. The minidisk is not accessible if the owner or any other user has a link to the minidisk in read or write status.

**WR**

Write access is desired. Only R/O access is allowed if the owner or any other user has a link to the minidisk in read or write status. Access is denied if any exclusive links exist.

**M**

Multiple access is desired. A write link is allowed to the minidisk unless another user already has write, stable or exclusive access to it, in which case, the minidisk is not accessible to you.

**MR**

Write or any exclusive access is allowed to the minidisk unless another user already has write access to it, in which case R/O access is allowed to the minidisk. Access is also denied if any exclusive links exist.

**MW**

Write access is allowed to the disk unconditionally, except for existing stable or exclusive links. Access is denied if any stable or exclusive links exist.

***suffix***

specifies an optional mode suffix to be directly appended to the access mode. The suffix letters can be combined as follows (note that while CP requires they be specified in this order, DirMaint allows them in any order and will reorder them appropriately):

1. S, E, or null (S and E are mutually exclusive), plus
2. D or null.

Therefore, the only valid combinations are: S, E, D, SD, or ED. This suffix is in turn concatenated with the mode, with no intervening blanks. For example, RS, RRE, and MRSD are all valid.

**Suffix****Definition****S**

Authorizes the virtual machine to use the LINK command stable access modes (SR, SW, SM) against the specified user's minidisk.

**E**

Authorizes the virtual machine to use the LINK command exclusive and stable access modes (ER, EW, SR, SW, SM) against the specified user's minidisk.

**D**

Tells CP that the device should not be defined when the virtual machine initially logs on or is autologged, but to defer doing so until an explicit LINK command is issued for that device.

***password***

defines the virtual machine owner's access password for the target minidisk. It must correspond to the mode of access that you are requesting. You may enter it as the last operand of the command as long as you take appropriate security precautions. It is recommended that you omit this operand and enter the password with display inhibited after the system requests it.

**DELETE**

specifies that you are voluntarily dropping the link.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa, PROMPT) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Use of the PROMPT prefix operand will result in command line prompting for a missing password.
3. For more information on the LINK directory statement, see [z/VM: CP Planning and Administration](#).
4. The owner becomes aware of your link the next time they issue the REVIEW operand and may decide to drop the link.
5. If the owner of the target minidisk has disabled their virtual machine for the DIRM LINK operand (by means of the DIRM USEROPTN LINKS DISABLE command), then you will not be able to establish a link

## LINK

to the minidisk even if you know the password; unless you have been specifically authorized to link to the device (by a DIRM AUTHLINK command) or you are authorized to act for ALL users. For more information, see [“AUTHLINK” on page 35](#).

6. The access password must be supplied but need not be correct if it is ALL or if the transaction originator is authorized to act for ALL users or you have been previously authorized to establish the link. For more information, see [“AUTHLINK” on page 35](#).
7. The maximum number of indirect links allowed can be set by the LINK\_MAX\_INDIRECT= setting in the CONFIG\* DATADVH. The default value is 50.

An indirect link is a link to a LINK statement in another directory entry which in turn links to a minidisk. Consider the following example:

If these statements are in the indicated directories:

```
USERA:
  MDISK 191 3380 1 10 MJHVOL ALL ALL ALL

USERB:
  LINK USERA 191 192 RR

USERC:
  LINK USERB 192 193 RR
```

USERA owns the actual minidisk statement as their 191 disk. USERB has a direct link to it as their 192 disk. USERC has an indirect link to USERA's disk as their 193 disk.

If the number of indirect links exceeds the current maximum allowed indirect link count, the link attempt is rejected with a message.

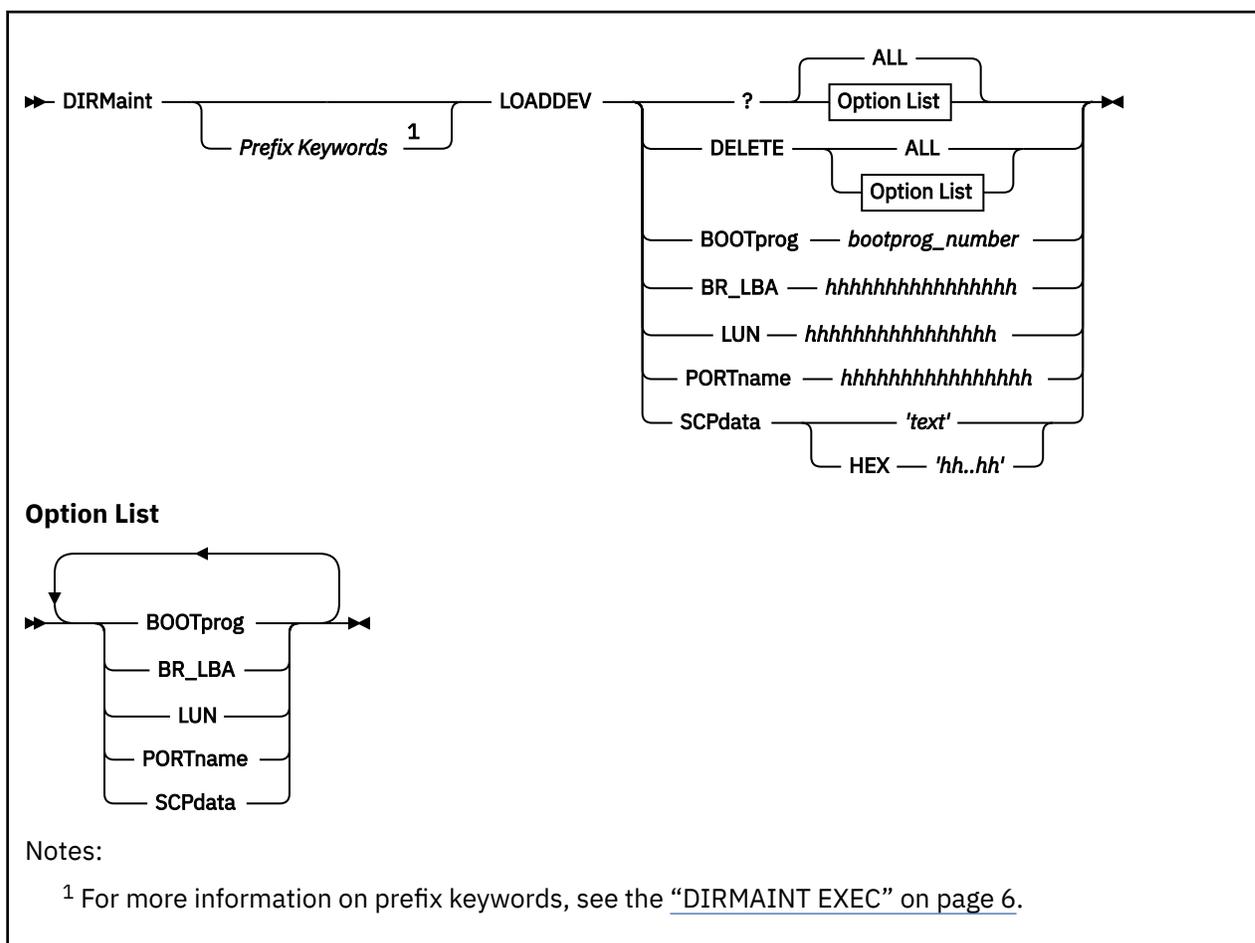
If the command handler is unable to follow the indirect chain to the root minidisk statement, the LINK attempt will be rejected with a message.

8. If the DIRM LINK threshold function is enabled, the LNKLMT= statement (located in the CONFIG\* DATADVH file) attempts to establish a threshold for failed link attempts. A file named DVHLINK ATTEMPT contains an entry for each user who has issued an invalid or incorrect DIRM LINK operand. Each entry includes the number of unsuccessful link attempts and the date the last unsuccessful attempt was made.

When the counter of invalid attempts for a user reaches the specified threshold, a message containing this information is sent to the user and the link attempt is rejected.

9. The DVHLINK ATTEMPT file can be administered using the DIRM ELINK operand.

## LOADDEV



### Purpose

Use the **LOADDEV** operand of the **DIRMAINT** command to identify the location of a program to be loaded as a result of an FCP List-Directed IPL.

### Operands

**?**

will query the statement or group of statements.

#### **DELETE**

will remove the statement or group of statements from the directory.

#### **BOOTprog**

The decimal value between 0 and 30 representing the script to IPL from.

#### **BR\_LBA**

The hexadecimal digits designating the logical-block address of the boot record relative to the beginning of the indicated partition of the FCP-I/O device. It must be a value from 0-FFFFFFFF FFFFFFFF. The default is 0.

#### **LUN**

The hexadecimal digits representing the 8-byte logical unit number of the FCP-I/O device. It must be a value from 0-FFFFFFFF FFFFFFFF. The default is 0.

**PORTname**

The hexadecimal digits designating the 8-byte fibre channel port name of the FCP-I/O device. It must be a value from 0-FFFFFFFF FFFFFFFF. There is no default.

**SCPDATA**

Designates information to be passed to the program which was loaded during list-directed IPL. If the program does not require any information, then this parameter is optional. Up to 4096 (4K) characters of data (text or hex) may be entered for SCPDATA. The actual number of input characters could be less 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 using the HEX option is 2048 (1/2 of the 4K character input limit). There is no default.

The quote marks around the information to be passed are required. See also usage note [“3” on page 148](#).

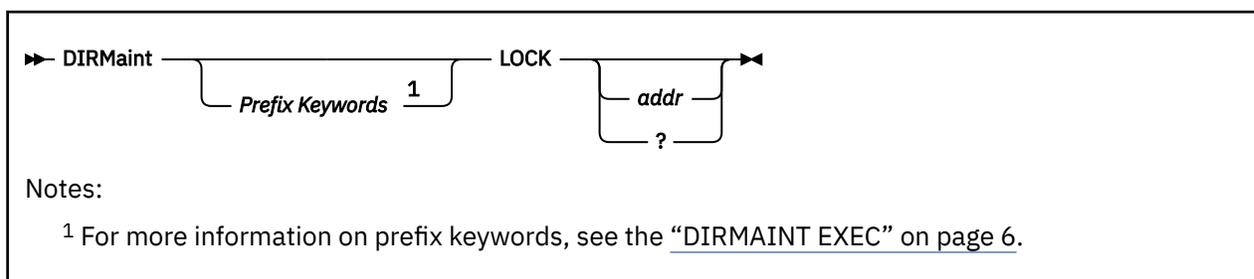
**HEX**

Indicates that the parameter value that follows is UTF-8 encoded hex (characters 0-F). If not specified, the value is assumed to be EBCDIC text (codepage 924).

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords do not apply to this command. They are accepted without error, but otherwise ignored.
2. If less than 16 hex digits are provided following the BR\_LBA, LUN, or PORTname keywords, leading zeros will be presumed.
3. When using the SCPDATA keyword, the information to be passed to CP for inclusion in the source directory can be specified in one of two ways:
  - Within a 'text' string, where *text* may include blanks or special characters. DirMaint will pass this string to CP exactly as specified. Blanks are not removed.
  - Within a HEX 'hh..hh' string, where *hh..hh* must be pairs of hex digits. Unlike above, special characters are not allowed in this string. DirMaint will allow blanks in this string, but will remove them before passing the statement to CP.
4. For more information on specifying quoted strings and the LOADDEV directory statement, see [z/VM: CP Planning and Administration](#).

## LOCK



### Purpose

Use the LOCK operand of the DIRMAINT command to temporarily prevent any user from making any changes to the directory entry associated with the FORuser and AT prefixes. The LOCK operand can be used to lock profile, user, identity and subconfiguration entries.

### Operands

#### *addr*

specifies the device address to lock.

#### ?

requests that the current status of the directory entry be returned.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The LOCK operand can be used whenever it is appropriate to inhibit you from making any changes directly to your own directory entry.
4. The LOCK operand will be invoked automatically whenever the GET operand is used unless the NOLOCK option is used with GET. Normally this is the only case that you would use it to lock a profile entry.
5. A lock request is refused with messages if a lock already exists for the directory entry, or if any minidisk device locks are currently enabled for the directory entry.
6. You cannot lock an entire directory entry if there are already any existing device locks for that entry. Similarly, you cannot lock a device if there is already an existing lock on the entire directory entry.

## LOGMSG

► DIRMaint — *Prefix Keywords* <sup>1</sup> — LOGmsg — *msgtext* ►

Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the LOGMSG operand of the DIRMAINT command to place a note into the history file that is maintained by the DIRMAINT virtual machine. A message that contains text from the note is issued to the console.

### Operands

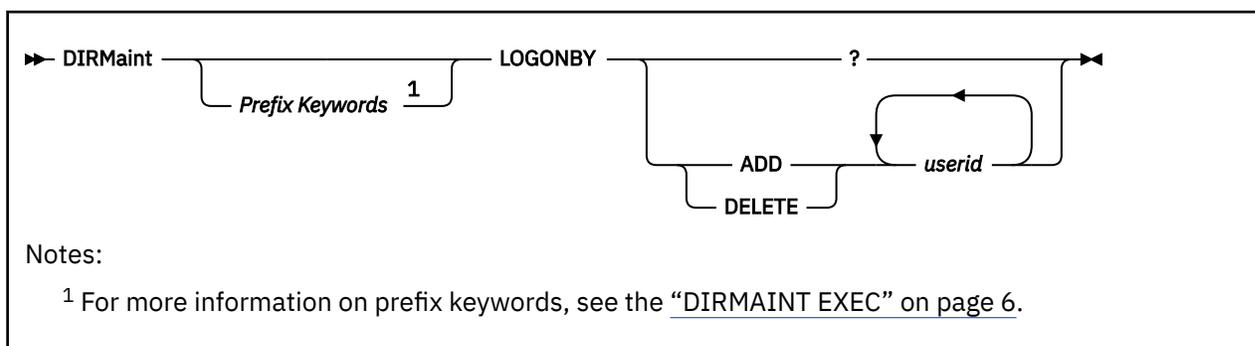
#### *msgtext*

is the message text that is to be recorded in the DirMaint history file.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. All transactions and other significant events are recorded by DirMaint. This operand can be used to place a note into this file, as a historical record. The history file is a daily log. Disposition of the current log at the end of a day is an installation option. The current history log can be retrieved at any time. For more information, see [“HISTORY”](#) on page 130.

## LOGONBY



## Purpose

Use the LOGONBY operand of the DIRMAINT command to add or delete users from the current LOGONBY directory statement in a profile, user, or identity entry. To determine the current users on the list, a query option is supported. The LOGONBY directory statement lists users that can use their own passwords to logon to different user IDs.

## Operands

?

indicates that a query request is being made.

**ADD**

**DELETE**

indicates that the listed user IDs are to be added to, or deleted from the list currently in the directory.

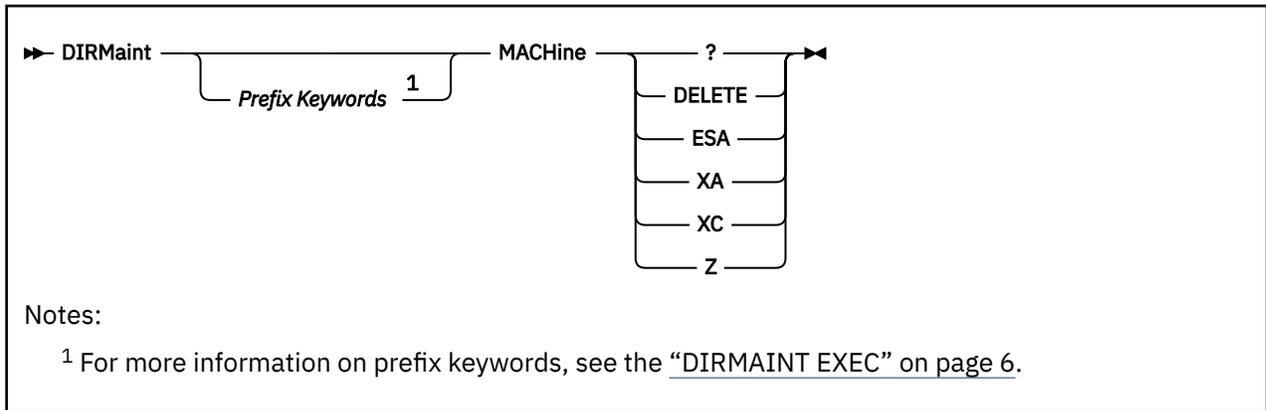
*userid*

a 1 to 8 character user ID that is to be added to or deleted from the LOGONBY directory statement.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the LOGONBY directory statement, see *z/VM: CP Planning and Administration*.
3. Attempts to delete a non-existent user ID will result in an error message and the command will be rejected.
4. Attempts to add a user ID that already exists will result in an error message and the command will be rejected.
5. If a LOGONBY directory entry does not exist, one will be created.
6. A maximum of eight users can exist on the LOGONBY directory statement in the user directory.
7. If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## MACHINE



### Purpose

Use the MACHINE operand of the DIRMAINT command to allow the general user to create or change a MACHINE statement in a profile, user, or identity entry. If a MACHINE statement does not exist in the directory entry, a new statement is created. If a MACHINE statement does exist, the machine type is changed as specified in the MACHINE operand.

### Operands

**?**

requests a query be done of the current MACHINE statement.

**DELETE**

requests that the MACHINE statement be deleted from the directory.

**ESA**

defines an ESA-mode virtual machine.

**XA**

defines an XA-mode virtual machine.

**XC**

defines an XC-mode virtual machine.

**Z**

defines a z/Architecture<sup>®</sup>-only virtual machine.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For a complete description of the MACHINE directory statement, see [z/VM: CP Planning and Administration](#).
3. For more information about the various modes of operation, see the CP SET command in the [z/VM: CP Commands and Utilities Reference](#).
4. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.
5. When a MACHINE statement is removed from a directory entry, the GLOBALOPTS machine definition will not become active until a DIRECTXA has been run against the full user directory file. To run

DIRECTXA against the full user directory file, issue a DIRMaint DIRECT command. (See [“DIRECT” on page 92.](#))

## MAIL



### Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

## Purpose

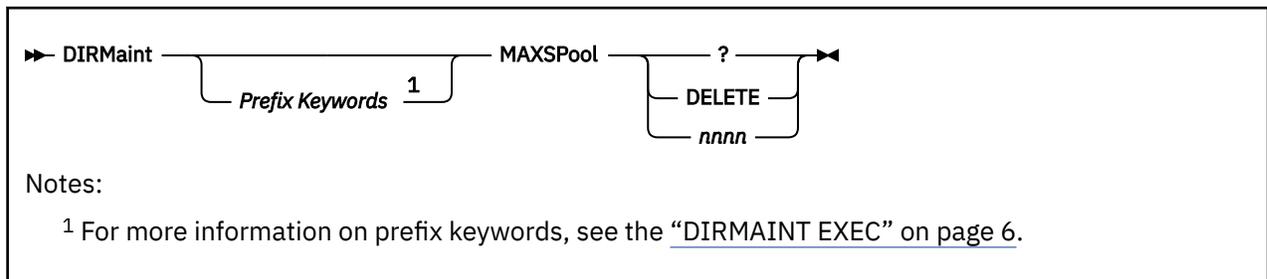
Use the MAIL operand of the DIRMAINT command to cause a specific file (DIRMAINT NEWMAIL A) to be forwarded to DirMaint and stored on DirMaint's user interface disk.

This operand is intended for use by system administrators who wish to forward mail to all DirMaint users. The mail file may contain DirMaint system information about new or changed function or installation policy. Mail recipients are advised to load mail from their virtual reader using RECEIVE to review the information.

## Usage Notes

1. You may use one or more of the prefix keywords (REQest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. This operand can be used to send a MAIL file to users of DirMaint. Actually, the MAIL file will be stored in the DIRMAINT virtual machine, and will be disseminated to users when they next invoke a DIRMAINT operand.
3. This is a mechanism whereby users can be notified of changes, new function, new policies, and so on. Once a MAIL file has been set up, users who subsequently invoke a DIRMAINT operand will automatically receive a copy of the MAIL file in their virtual card readers. Any given user with a R/W A-minidisk or directory will be sent only one copy of the MAIL file.
4. Prior to invoking this operand, you, as the system administrator, must have used an editor, in your own virtual machine, to create the file DIRMAINT NEWMAIL on your A-minidisk.

# MAXSPool



## Purpose

Use the MAXSPool operand of the DIRMAINT command to add, delete, change, or query a SPOOLFILE directory control statement in a directory entry.

## Operands

**?**

specifies a request to query the current SPOOLFILE statement.

### DELETE

specifies a request to delete the SPOOLFILE statement from the directory entry.

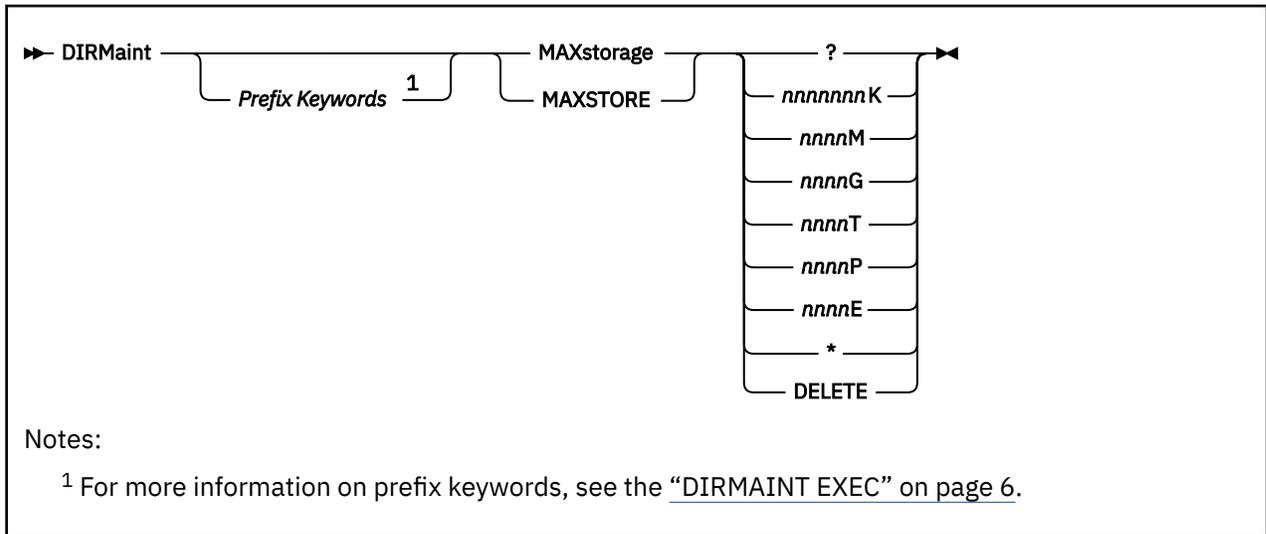
### nnnn

specifies the number of spool files that the subject user is allowed to have. This number can range from 1 to 9999.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOSys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For a complete description of the SPOOLFILE directory statement, see [z/VM: CP Planning and Administration](#).
3. If no SPOOLFILE MAXSPool statements are specified, the CP default of 9999 spool files is in effect.
4. If a SPOOLFILE statement does not exist in the directory definition and DELETE is not specified, a new statement is created. If such a statement exists and DELETE is specified, the statement is removed. If a SPOOLFILE statement exists and DELETE is not specified, the supplied value replaces whatever was previously on the record after appropriate validation is complete.

## MAXSTORAGE, MAXSTORE



### Purpose

Use the MAXSTOR operand of the DIRMAINT command to change the maximum virtual storage size in a directory entry.

### Operands

**?**

specifies that this transaction is a query to determine the current maximum virtual storage size for this entry.

**nnnnnnnK**

specifies the maximum value allowed in kilobytes.

**nnnnM**

specifies the maximum value allowed in megabytes.

**nnnnG**

specifies the maximum value allowed in gigabytes.

**nnnnT**

specifies the maximum value allowed in terabytes.

**nnnnP**

specifies the maximum value allowed in petabytes.

**nnnnE**

specifies the maximum value allowed in exabytes.

**\***

when used within a:

- User or identity directory entry, specifies that the value specified within the included profile entry will be used if present, or the CP default value of 1M will be used if there is no MAXSTORAGE value within the included profile; and
- Profile directory entry, specifies that the CP default value of 1M will be used unless the including user or identity entry specifies an overriding MAXSTORAGE value.

**DELETE**

specifies that the MAXSTORAGE statement will be deleted, resulting in use of the:

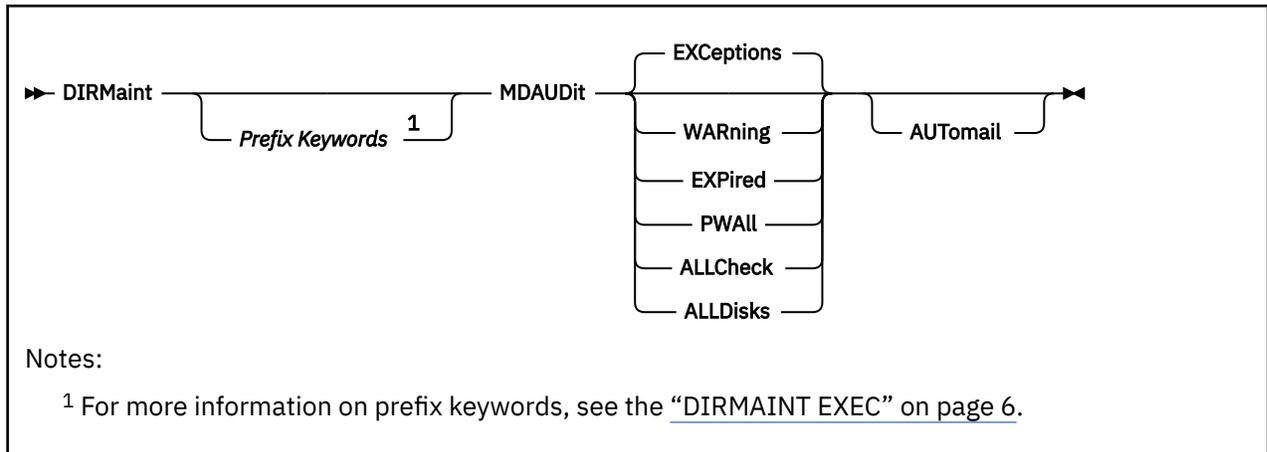
- Value specified within an included profile if deleted from a user or identity directory entry.

- CP default value of 1M when deleted from a profile entry, unless the including user or identity entry specifies an overriding MAXSTORAGE value.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the USER and IDENTITY directory statements, see [z/VM: CP Planning and Administration](#).
3. The value is placed into the directory as provided once it has been validated for correct format. Any required rounding is done by CP.
4. If the directory definition of the user ID being modified does not currently have a default storage size defined, a default logon storage size of 256KB is assigned to the user ID. This value becomes the minimum that can be specified for the maximum virtual storage size.
5. The values supplied must be whole numbers. Values such as 4096.8K or 6.5M are not supported.
6. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.
7. The suffix K, M, G, T, P or E is required.
8. A query request will report the default values for the default storage size and maximum storage size if they are not specified on the USER or IDENTITY statement.
9. The value specified with this command will be added to the corresponding field in the USER or IDENTITY statement.

## MDAUDIT



### Purpose

Use the MDAUDIT operand of the DIRMAINT command to check for compliance with established policy. If an installation is using the MDISK password monitoring function.

### Operands

#### EXCeptions

all disks showing an exception (password of ALL, expired, or in the warning period) are presented. This is the default.

#### WARning

all minidisks that are considered to be expired or in the warning period are presented.

#### EXPIred

only disks with expired passwords are presented in the report.

#### PWAll

produces a report that is sent to the invoker as a spool file listing all instances of the password ALL. Two circumstances are recognized:

1. ALL used as a READ password. This is a concern only from a security standpoint, and is usually done deliberately to effectively make the contents of a disk *public*.
2. ALL used for a WRITE or MULTIPLE access password. If this is done by a general user, it is probably a mistake. A serious data integrity problem can develop, since any other user can write and erase files on such a disk.

#### ALLCheck

all disks showing an exception (password of ALL, expired, or in the warning period) are presented. This option is synonymous with option EXCEPTIONS.

#### ALLDisks

all disks that have minidisk passwords are presented. Even disks without any exceptions are reported.

#### AUTomail

sends electronic mail directly to the users who own minidisks for which exceptions are detected. This *form letter* is described under Usage Notes “5” on page 160, “6” on page 161, “7” on page 161, and “8” on page 161.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but

each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.

- MDISK Password Monitoring is a built-in function that is inactive by default. Installations that want to ensure that MDISK passwords are changed periodically may define the appropriate frequency in the DirMaint configuration file. For example,

```
MDPW_INTERVAL= ww ll
```

Where:

**ww**

specifies the time (in days) until the user is considered to be in the WARNING period.

**ll**

specifies the time (in days) until the user is considered to have expired passwords.

The values for these intervals should be in the range:

```
0 <= ww <= ll <= 366
```

If this entry is not found, if both values are not provided or the values are in an invalid format, the MDAUDIT command is rejected with a message.

DirMaint time stamps the password changes on a minidisk basis. Three time stamps are maintained; one each for READ, WRITE, and MULTI.

- Once minidisk password time stamping is in effect, you may begin using the MDAUDIT function to detect instances of non-compliance with your defined policy. It will only report exceptions as detected. Installations will develop their own appropriate means of dealing with failure to adhere to password management policy.
- The normal output of the MDAUDIT function is a spool file returned to the invoker, which should be loaded from the virtual reader by the CMS RECEIVE command. This file contains one record for each minidisk for which some exception has been noted. The passed parameters affect the content of the report. This file is intended to be visually readable, as well as machine readable. Since it is expected that most installations will create some sort of automatic procedure to process this file, column headings are not included. To aide in the development of an application, all fields are filled in with one or more — characters if they are not meaningful to the exception report. The returned file has the following fields in the specified order:

**Field Name**

**Meaning and value**

**User ID**

The user ID that owns the mdisk device.

**SysAffin**

The System Affinity of the device.

**Ccuu**

The address of the device.

**Rtype**

The type of READ exception associated with this device.

Valid exceptions for READ, WRITE, and MULTI include:

**Exception**

**Meaning**

**NONE**

No exception noted for this password.

**WARN**

This password is in the warning period.

**\*EXP**

This password is expired.

**\*ALL**

This password was found to be set to ALL.

**Rcd**

The READ change date. With the format

```
yyyy/mm/dd
```

**Rdsc**

The number of days since the READ password was changed.

**Rdod**

The number of days a READ password change is overdue, or the string --- if the password change is not overdue.

**Rdtcd**

The number of days until a READ password change is due, or the string --- if the password is currently overdue.

**Wtype**

The type of WRITE exception associated with this device.

See READ above for detailed meanings of exception types.

**Wcd**

The WRITE change date. With the format

```
yyyy/mm/dd
```

**Wdsc**

The number of days since the WRITE password was changed.

**Wdod**

The number of days a WRITE password change is overdue, or the string --- if the password change is not overdue.

**Wdtcd**

The number of days until a WRITE password change is due, or the string --- if the password is currently overdue.

**Mtype**

The type of MULTI exception associated with this device.

See READ above for detailed meanings of exception types.

**Mcd**

The MULTI change date. With the format

```
yyyy/mm/dd
```

**Mdsc**

The number of days since the MULTI password was changed.

**Mdod**

The number of days a MULTI password change is overdue, or the string --- if the password change is not overdue.

**Mdtcd**

The number of days until a MULTI password change is due, or the string --- if the password is currently overdue.

**Rall**

This field will contain an X if the READ password is ALL, it will contain a - if it is not.

**Wall**

This field will contain an X if the WRITE password is ALL, it will contain a - if it is not.

5. When the AUTOMAIL parameter is specified, a form letter is sent to users with detected exceptions as listed in the *Type of Exception* column.

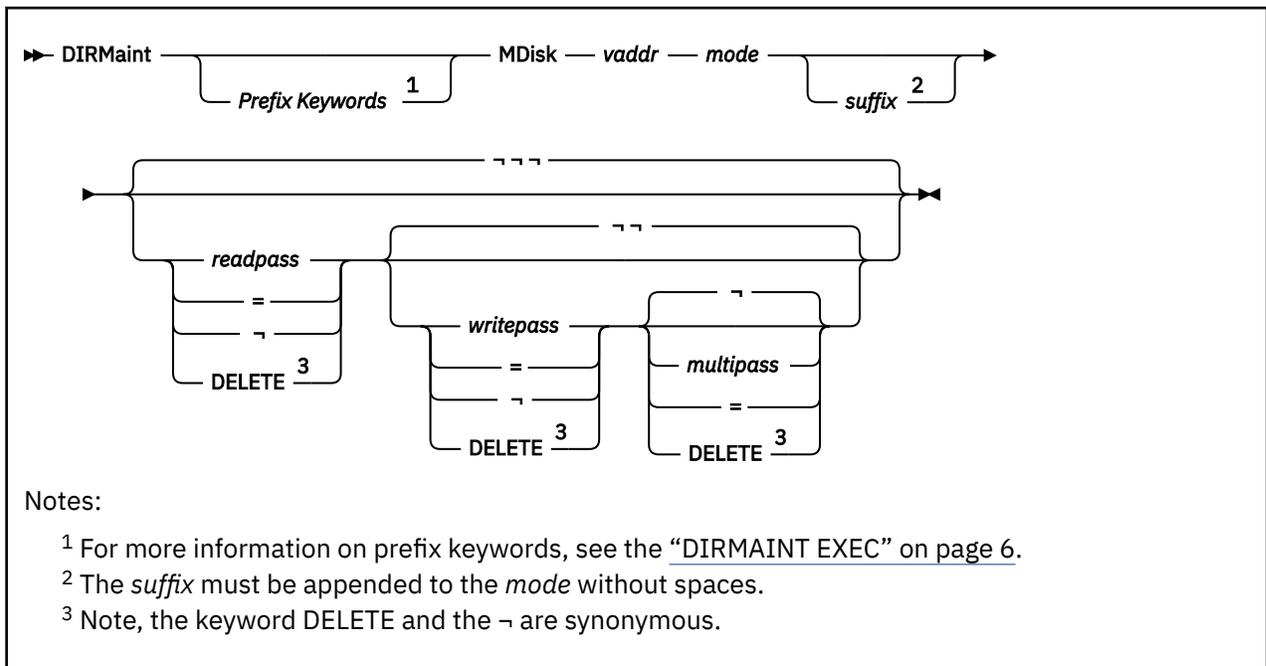
6. DirMaint is shipped with a default form letter to be sent to offenders when AUTOMAIL is used. The first two lines of the output are automatically generated by the MDAUDIT processor and contain the user ID of the individual receiving the letter and the date on which the MDAUDIT program was run. The form letter is drawn from the AUTOMAIL DATADVH file. This form letter should be customized to reflect the correct administrator name and phone extension. When altering this file do not alter the prefix tags that start each line of the report. The MDAUDIT processor uses these to assign position within the final output file.
7. Leap years are not accounted for during date subtraction. February is assumed to always have 28 days, regardless of year.
8. The report sent to the invoker is terminated with a line containing:

```
END-REPORT
```

9. If a value exceeds 999, it is set to 999 in the report.
10. AUTOMAIL sends the generated form letter to

```
userid AT *
```

## MDISK



### Purpose

Use the MDISK operand of the DIRMAINT command to allow you to change the access mode and the access passwords for a minidisk in a user, identity, or subconfiguration entry.

### Operands

#### *vaddr*

is the virtual device address of the minidisk being changed.

#### *mode*

specifies the user's mode of access at logon. If the minidisk owner allows other users access to the minidisk by establishing access passwords, then the access conditions may be modified by such links. Valid access modes are:

#### Mode

##### Definition

=

The user is not changing the access mode or suffix. This request only adds, deletes, or modifies access passwords.

R

Read-only (R/O) access is desired. The access is not given if any other user has a link to the disk in write or exclusive status.

RR

Read-only (R/O) access is desired, even if another user has a link to the disk in write status. Access is denied if any user has an exclusive link.

W

Write access is desired. The disk is not accessible if any other user has a link to the disk in read or write status.

WR

Write access is desired, but read-only access is acceptable if any other user has a link to the disk in read or write status. Access is denied if any user has an exclusive link.

**M**

Multiple access is desired. A write link to the disk is given unless another user already has write or any exclusive access to it; in which case, the disk is not accessible.

**MR**

Write access to the minidisk is desired unless another user already has write or any exclusive access to it. In this case, read-only access to the minidisk is allowed. Access is denied if any user has an exclusive link.

**MW**

Unconditional write access to the disk is desired. Access is denied if any user has an exclusive or stable link.

**suffix**

specifies an optional mode suffix to be directly appended to the access mode. The suffix letters can be combined as follows (note that while CP requires they be specified in this order, DirMaint allows them in any order and will reorder them appropriately):

1. V or null, plus
2. S, E, or null (S and E are mutually exclusive), plus
3. D or null.

Therefore, the only valid combinations are: V, S, E, D, VS, VE, VD, VSD, VED, SD, or ED. This suffix is in turn concatenated with the mode, with no intervening blanks. For example, RV, RRS, WVE, and MVSD are all valid.

**Suffix****Definition****V**

Tells CP to use its virtual reserve/release support in the I/O operations for the minidisk.

**S**

Authorizes the virtual machine to use the LINK command stable access modes (SR, SW, SM) against the specified user's minidisk.

**E**

Authorizes the virtual machine to use the LINK command exclusive and stable access modes (ER, EW, SR, SW, SM) against the specified user's minidisk.

**D**

Tells CP that the device should not be defined when the virtual machine initially logs on or is autologged, but to defer doing so until an explicit LINK command is issued for that device.

**readpass**

if you specify a read password, then any user knowing it can establish a read-only link to your minidisk. The password must be specified as a character string of 1 to 8 characters that must not contain embedded blanks. The character string ALL permits any user to establish a read-only link without knowing the correct password.

**writepass**

you cannot specify this option unless you have a read password. If a user specifies a write password, then anyone else knowing it can acquire write access to the disk. The password must be specified as described in the *readpass* definition. The character string ALL is **not** recommended for use as a write access password.

**multipass**

you cannot specify this option unless you have both read and write passwords. If the user specifies a multiple password, then anyone else knowing it can acquire multiple-write access to the disk. The character string ALL is **not** recommended for use as a multiple-write access password.

=

indicates that the specified access password is not to be changed.

**DELETE**

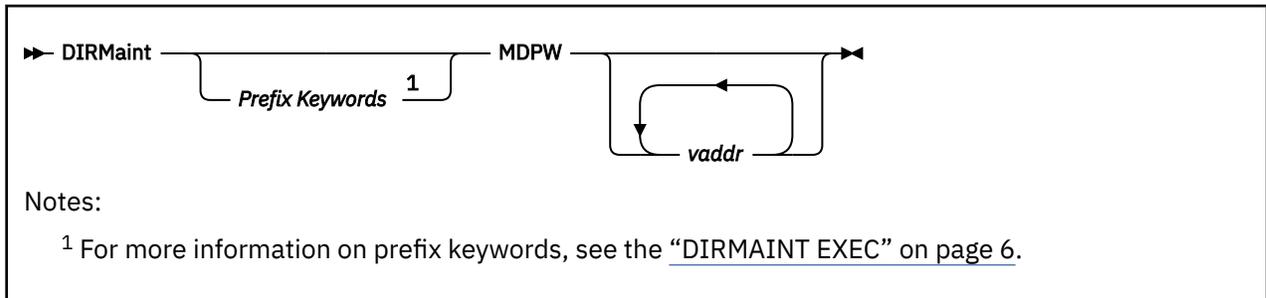
-

indicates that the specified access password is to be deleted.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa, PROMPT) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Use of the PROMPT prefix operand will result in command line prompting for a missing password.
3. For more information on the MDISK directory statement, see [\*z/VM: CP Planning and Administration\*](#).
4. Most users do not want to establish a multiple-access password. You must understand the implications of establishing a multiple-access password before you do it.
5. If a password is removed, then any dependent passwords (passwords to the right of the removed password) are also removed.
6. Passwords are subject to acceptance by the user exit facility.
7. If you enter passwords directly on the command line, you should follow installation guidelines for security. It is better, from a security standpoint, not to enter access passwords on the command line. If they are omitted and the system prompts you for them, they are entered in the *display inhibit* mode. To be prompted for the passwords, specify the PROMPT prefix keyword. For example:  
  
DIRM PROMPT MDISK 191 MR  
  
Enter either one, two or all three passwords at the prompt.
8. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## MDPW



### Purpose

Use the MDPW operand of the DIRMAINT command to obtain the access passwords for one or all of the minidisks in a user, identity, or subconfiguration entry. These passwords are part of the response message(s) written to your terminal. A message is issued for each selected minidisk, even if it does not have an access password. A header message is included in the output only if your installation is exploiting minidisk password monitoring.

### Operands

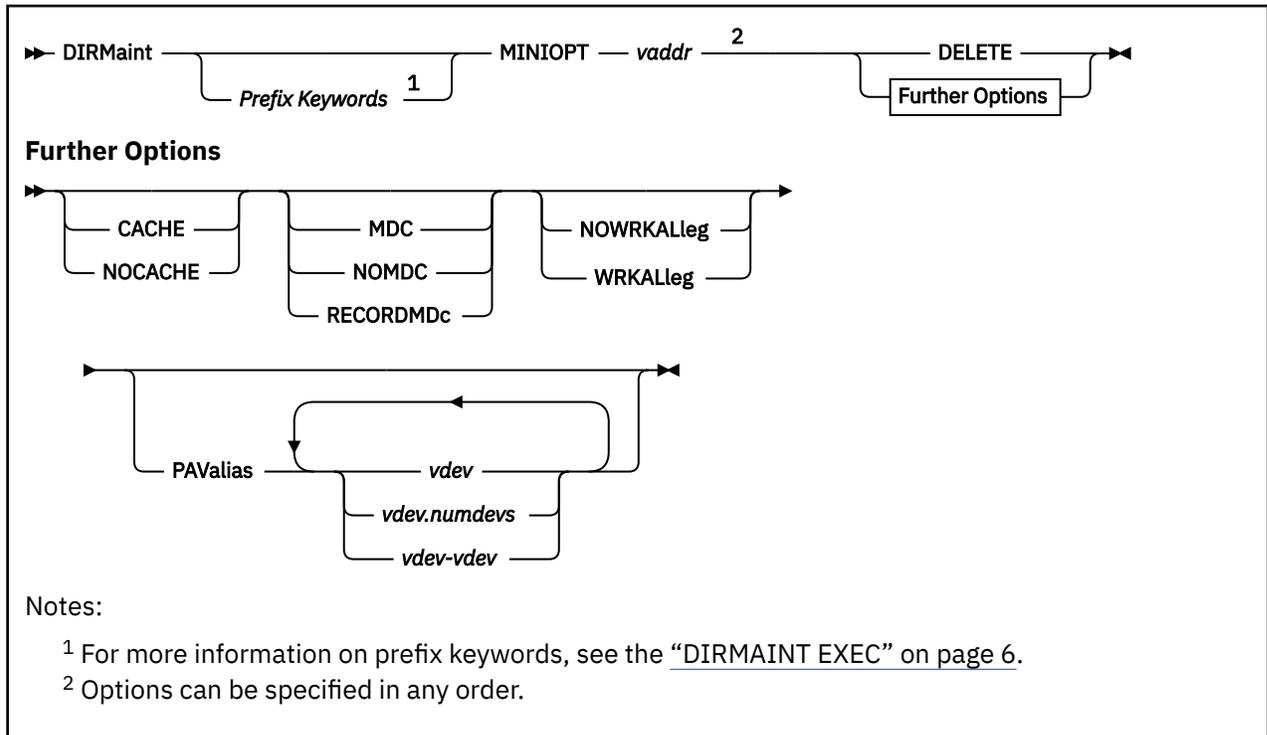
#### *vaddr*

is the virtual device address of a particular minidisk. If this parameter is not entered, all minidisks will be listed.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa, MENU) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. Your installation may have an established policy regarding passwords and security precautions. For example, many installations consider passwords to be confidential. If you are not working in a secure area, be prepared to safeguard or carefully dispose of the resulting output. If your virtual machine is in a *not receiving* state, the messages will not be recorded in your message log.

## MINIOPT



### Purpose

Use the MINIOPT operand of the DIRMAINT command to add, replace, or delete an existing MINIOPT statement in a user, identity, or subconfiguration entry.

### Operands

#### DELETE

means that the specified MINIOPT directory control statement should be deleted.

#### vaddr

is the virtual address of the MDISK to which the new or changed MINIOPT statement should apply.

#### CACHE

#### NOCACHE

indicates if the DASD will have access to the cache, or if CP will force I/O for the minidisk to bypass the cache.

#### MDC

#### NOMDC

#### RECORDMDC

indicates whether the minidisk will use full track minidisk cache, record level minidisk cache, or no cache.



**Attention:** The RECORDMDC option should be used only when directed to do so by IBM support personnel. RECORDMDC is for use only when DASD cache as well main storage and expanded storage minidisk cache are all being over-whelmed by excessive unreferenced records being read by the full track cache support. For more information, see [z/VM: CP Planning and Administration](#).

#### WRKALleg

#### NOWRKALleg

indicates whether working allegiance will be simulated on the minidisk.

**PAValias**

defines one or more alias parallel access volumes for the non-fullpack minidisk base parallel access volume specified in *vaddr*. One virtual device is created for each specified *vdev*.

**vdev****vdev.numdevs****vdev-vdev**

specifies a single virtual device (*vdev*), a virtual device combined with a decimal range count (*vdev.numdevs*), or a range of virtual devices (*vdev-vdev*) to be added or updated on the MINIOPT statement.

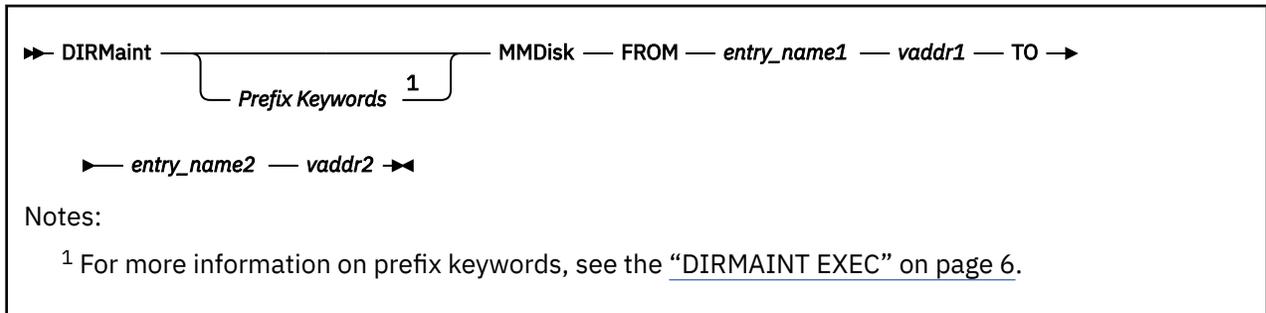
**Note:**

1. The device numbers entered must be hexadecimal numbers between X'0000' and X'FFFF'.
2. *numdevs* must be a decimal number.
3. Multiple device specifications must not overlap.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Not every option is valid in every system environment. For more information on the MINIOPT directory statement, see *z/VM: CP Planning and Administration*.
3. A MINIOPT statement is a logical extension to an MDISK statement in the directory. This statement is used to specify whether CP should use the caching function for I/O to that minidisk.
4. When the DIRMAINT command is issued with the MINIOPT operand (add or replace transaction) for a device which already has an associated MINIOPT statement, the existing MINIOPT statement is replaced. The new MINIOPT statement contains only the options specified for the MINIOPT operand.

## MMDISK



### Purpose

Use the MMDISK operand of the DIRMAINT command to copy the extents of an existing permanent minidisk in a user, identity, or subconfiguration entry to an existing permanent minidisk in another user, identity, or subconfiguration entry.

The extents (device type, starting cylinder, number of cylinders, and volume serial) that define the source minidisk are copied to the target. No data movement occurs. After the MMDISK operand is processed, both the source and the target minidisk point to the same location on DASD. None of the fields of the source minidisk statement are changed. None of the fields of the target minidisk statement are changed except the extent fields. For example, the default link mode and password of this minidisk stay the same. Associated MINIOPT or DASDOPT statements are not moved.

### Operands

#### FROM

is a required keyword

#### *entry\_name1*

is the name of the directory entry where the source minidisk is defined.

#### *vaddr1*

is the virtual device address of the source minidisk.

#### TO

is a required keyword.

#### *entry\_name2*

is the name of the directory entry where the target minidisk is defined.

#### *vaddr2*

is the virtual device address of the target minidisk.

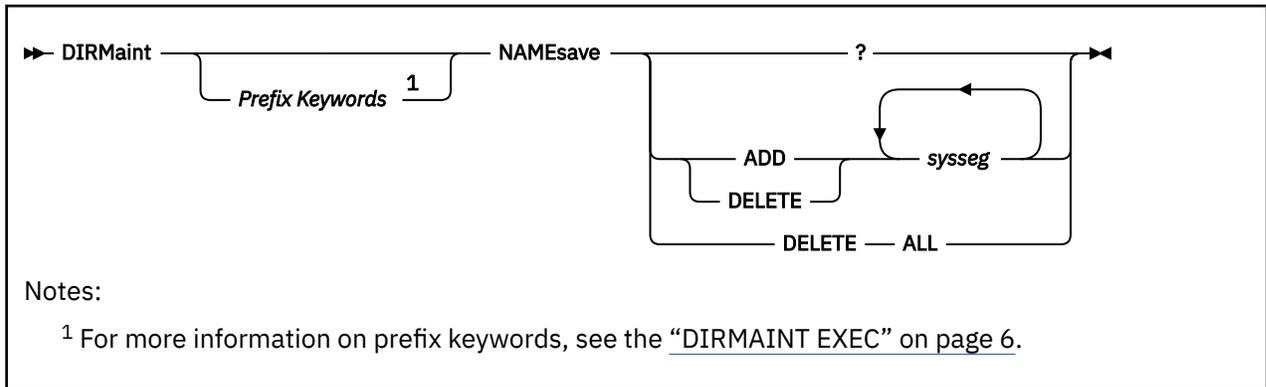
### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. A profile entry cannot be specified for *entry\_name1* or *entry\_name2*.
3. A specification of *entry\_name1* = *entry\_name2* when *vaddr1* = *vaddr2* is prohibited.
4. If there are no other minidisks in the system that occupy the same extent as that occupied by the old target minidisk, DirMaint will consider that space as available for allocation to another minidisk (through either ADD, AMDISK, CMDISK or RMDISK).

**Note:** This has not been formatted, even if DISK\_CLEANUP= YES is specified in the CONFIG\* DATADVH file.

5. A DEVNO, T-DISK or a V-DISK MDISK can be the source or the target of a MMDISK operand.
6. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.
7. If the AT prefix is specified for a subconfiguration entry when the MMDISK operand is issued, the AT keyword value will be used to select the DATAMOVE machine configured at the specified node. In this respect, the FORuser prefix keyword must specify the subconfiguration entry name in order to modify a subconfiguration entry with the MMDISK operand.

## NAMESAVE



### Purpose

Use the NAMESAVE operand of the DIRMAINT command to add, delete, or query a system segment name from an existing NAMESAVE statement for a specified profile, user, or identity entry.

### Operands

**?**

requests a query of the current NAMESAVE directory entry.

#### ADD

specifies that the *sysseg* is to be added to an existing NAMESAVE statement, or a new statement is to be created.

#### DELETE

##### DELETE ALL

specifies that the *sysseg* is to be deleted from the NAMESAVE statement. If the ALL keyword is used the entire NAMESAVE statement is deleted.

#### *sysseg*

specifies a 1 to 8 character name of the restricted named saved system, or a discontinuous saved segment that the virtual machine is authorized to access.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the NAMESAVE directory statement, see [z/VM: CP Planning and Administration](#).
3. A *DIRM NAMESAVE ADD sysseg* may result in the creation of a NAMESAVE statement if there are no existing NAMESAVE statements in the directory entry, or no more room on an existing NAMESAVE statement.
4. A *DIRM NAMESAVE DELETE sysseg* may result in the deletion of a NAMESAVE statement if there are no more *syssegs* on the existing NAMESAVE statement.
5. Specifying DELETE for a *sysseg* name not on a NAMESAVE statement will generate an error message.
6. Specifying ADD for a *sysseg* name already on a NAMESAVE statement will also generate an error message.

## NEEDPASS



### Purpose

Use the NEEDPASS operand of the DIRMAINT command to activate and deactivate password prompting for DirMaint commands for the issuing user ID when the DirMaint configuration option NEEDPASS\_RESPECT is set to YES.

### Operands

**?**

allows you to query your current NEEDPASS setting. You will receive back a numerical value, where 0 = NO (OFF) and 1 = YES (ON).

**YES**

specifies that the user is to be prompted for a password when a command is entered.

**NO**

specifies that the user is not to be prompted for a password when a command is entered.

### Usage Notes

1. All prefix keywords are ignored. They are accepted without error, but otherwise ignored.
2. The NEEDPASS command is not valid in a BATCH command.
3. When you issue a DIRM NEEDPASS command and the DirMaint configuration option NEEDPASS\_RESPECT is set to YES, you will always be prompted for a password, regardless of your global variable settings or the DVHOPT entry in the directory entry settings. If a password is entered in response to the prompt, the user's global variable will be updated, regardless of whether the password is valid or not.
4. If DIRM NEEDPASS ? shows that the values of the global variable on the users machine and the DVHOPT entry in the directory entry settings differ, issue another DIRM NEEDPASS command to set them to the correct value and supply the correct password when prompted.
5. If your DIRMAINT transaction fails with message DVHREQ2286E when you are not prompted to enter a password, there is a mismatch between the NEEDPASS value in your directory entry and your LASTING GLOBALV, issue another DIRM NEEDPASS NO command to set them to the correct value. However, if your NEEDPASS NO request does not end with message DVHREQ2289I, correct your error and issue another DIRM NEEDPASS NO command to set them to the correct value.

## NEWS

---

**Notes:**

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

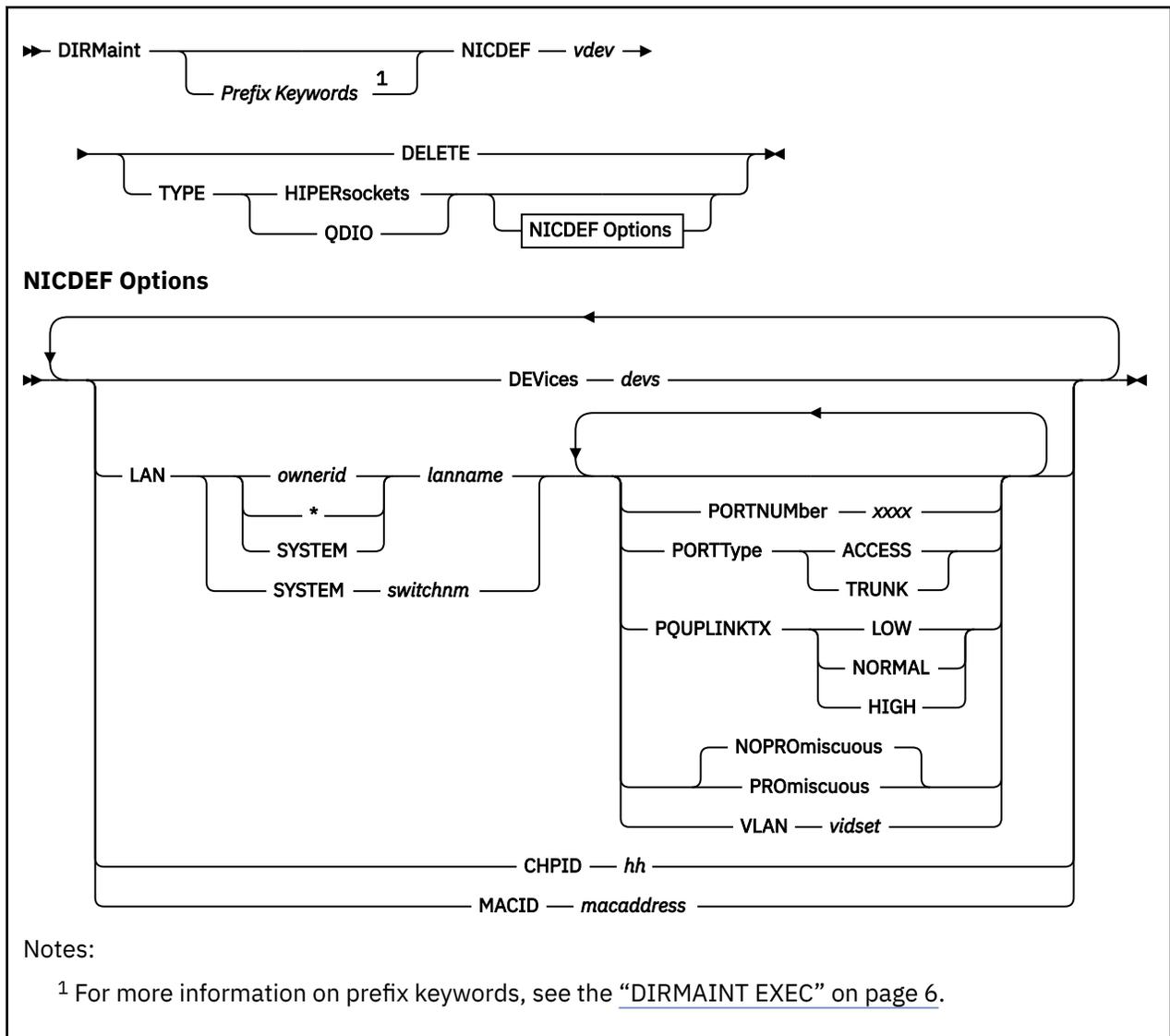
### Purpose

Use the NEWS operand of the DIRMAINT command to retrieve the current news file from the DIRMAINT server.

### Usage Notes

You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.

## NICDEF



### Purpose

Use the NICDEF operand of the DIRMAINT command to create a virtual HiperSockets or QDIO adapter in the virtual machine, and to optionally connect it to a VM LAN or z/VM Virtual Switch.

### Operands

#### DELETE

will remove the statement or group of statements from the directory.

#### vdev

is the base (or first) device address in a series of virtual I/O devices that belong to the same unit.

#### TYPE

specifies the type of NIC adapter to be created, specifically the hardware and protocol that the adapter will emulate. Note that TYPE is a required keyword and it *must* be the first keyword specified. If LAN is also specified, an attempt will be made to couple the adapter to the specified *ownerid lanname*.

The values that may be specified for TYPE are as follows:

**HIPERsockets**

defines this adapter as a simulated HiperSockets NIC. This adapter will function like a HiperSockets internal adapter (model 1732-05). A HiperSockets NIC can function without a guest LAN connection, or it can be coupled to a HiperSockets guest LAN. An error will result 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 an OSA Direct Express® (QDIO) adapter (model 1731-01). A QDIO NIC is functional only when it is coupled to a QDIO guest LAN or to a QDIO, IEDN, or INMN virtual switch. A QDIO adapter can be coupled to an IEDN or INMN virtual switch only when the owning user ID is authorized by the system administrator, by specifying OSDSIM on the SET VSWITCH command.

**DEVICES *devs***

is the number (decimal) of virtual I/O devices to be created for a simulated Network Interface Card (NIC). This number is evaluated during LOGON processing. For a simulated HiperSockets adapter, *devs* must be a decimal value between 3 and 3,072 (inclusive). For a simulated QDIO, IEDN, or INMN adapter, *devs* must be a decimal value between 3 and 240 (inclusive). If the *devs* value is omitted, the default for either HiperSockets, QDIO, IEDN, or INMN is three (3) devices.

**LAN *ownerid*/\**lanname***

identifies a virtual LAN segment for an immediate connection to the Network Interface Card (NIC). When *ownerid* is specified as an asterisk (\*), it is resolved as the user ID of the current virtual machine. When the LAN operand is omitted, the adapter is left in the default (uncoupled) state. When LAN *ownerid lanname* is identified in this statement or another with the same *vdev*, the adapter is connected to the designated virtual LAN segment automatically.

*Ownerid* may be specified as SYSTEM, indicating the virtual LAN segment is system-owned.

**LAN SYSTEM *switchnm***

identifies a z/VM Virtual Switch for an immediate connection to the Network Interface Card (NIC). When the **LAN** operand is omitted, the adapter is left in the default (uncoupled) state. When **LAN SYSTEM *switchnm*** is identified in this statement or another with the same *vdev*, the adapter is connected to the designated z/VM Virtual Switch automatically.

**LAN SYSTEM *switchnm*** is only valid for QDIO adapters.

**CHPID *hh***

is a two digit hexadecimal number that represents the Channel Path ID (CHPID) number to be allocated in the virtual machine I/O configuration for this adapter. If **CHPID** is omitted, an available CHPID is automatically assigned to this adapter. This option is required when a HiperSockets adapter is being created for a z/OS® guest because z/OS configurations require a predictable CHPID number. During LOGON, CP attempts to use the specified CHPID number. If the specified CHPID number is already in use, this adapter is not defined. To correct this situation, you must eliminate the conflicting device or select a different CHPID.

**MACID *macaddress***

is a unique identifier (up to six hexadecimal digits) used as part of the adapter MAC address for a QDIO or HiperSockets type NIC adapter. A MAC address for a device on an IEDN or INMN network can be assigned only by the Unified Resource Manager. Therefore, a user-defined MAC address is *not* allowed on these networks. Specifying a MACID for a type IEDN or INMN adapter would prevent the adapter from being added to the virtual I/O configuration of a guest.

During LOGON, your MACID (three bytes) is appended to the system MACPREFIX or USERPREFIX (three bytes) to form a unique MAC address for this adapter. If MACID is omitted from this definition, CP will generate a unique identifier for this adapter. If the specified MACID is already in use, this adapter will not be defined. To correct this situation, you must eliminate the conflicting device or select a different MACID.

**PORTNUMBER *portnum***

is the network-specific port number (a decimal number in the range 1-2048) to be used when this virtual NIC is connected to the NICDEF LAN.

**PORTType ACCESS|TRUNK**

is the network-specific port type to determine whether VLAN tags should be visible to the guest.

**ACCESS**

defines a connection that exchanges untagged frames with the guest. PORTTYPE ACCESS is only valid when the interface is configured for a single VLAN.

CP adds (or removes) VLAN tags as necessary.

**TRUNK**

defines a connection that exchanges tagged frames with the guest. PORTTYPE TRUNK is the only valid option when the interface is configured for multiple VLAN groups.

**PQUPLINKTX LOW | NORMAL | HIGH**

For a virtual switch with priority queuing enabled, PQUPLINKTX sets the priority for all packets sent from a NIC's network connection to an external network. If PQUPLINKTX is not specified, all outbound traffic to the external network is sent at a NORMAL priority on the virtual switch uplink port. If PQUPLINKTX is configured for a virtual switch without priority queuing enabled, the setting is saved and used if priority queuing is enabled at a later time.

For a HiperSockets type NIC, the PQUPLINKTX operand is ignored.

**LOW**

specifies that outbound traffic to the external network is sent at a low priority. This traffic uses the low priority queue, which is serviced less frequently than the normal and high priority queues.

**NORMAL**

specifies that outbound traffic to the external network is sent at a normal priority. This traffic uses 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 is sent at a high priority. This traffic uses the high priority queue, which is serviced more frequently than the normal or low priority queues.

**PROMISCUOUS | NOPROMISCUOUS**

when PROMISCUOUS is specified, the guest is authorized to enable promiscuous mode (allowing this interface to receive a copy of every network packet on the simulated LAN segment).

**VLAN vidset**

defines the VLAN ID set. The *vidset* may be a simple VLAN ID (for example, "VLAN 1"), a VLAN range (for example, "VLAN 10-19"), or a complex set (for example, "VLAN 1 10-19 100-109").

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords do not apply to this command. They are accepted without error, but otherwise ignored.
2. If all of the parameters are needed for a given device definition, then the resulting directory statement will not fit within a single directory record. The statement must be completely specified, however, within a single DirMaint request. DirMaint will therefore split the statement and continue it onto subsequent records by repeating the statement name and device address, followed by the additional parameters, as needed. Such continuation records will be consecutive. Intervening comments and blank lines are not supported.
3. When adding a new directory entry containing NICDEF statements, or when using a DIRM REPLACE command to add a NICDEF statement to a directory entry, note that the NICDEF statements for a given device address must be consecutive records with no intervening comment lines or blank records.
4. The LAN operand is required when a virtual switch-specific operand (PORTNUMBER, PORTTYPE, VLAN, PQUPLINKTX, PROMISCUOUS or NOPROMISCUOUS) is specified. Furthermore, the use of a virtual switch-specific operand restricts the virtual NIC to the designated network. When the virtual

## NICDEF

NIC is configured for a specific network, the CP COUPLE command does not allow a connection to any other network.

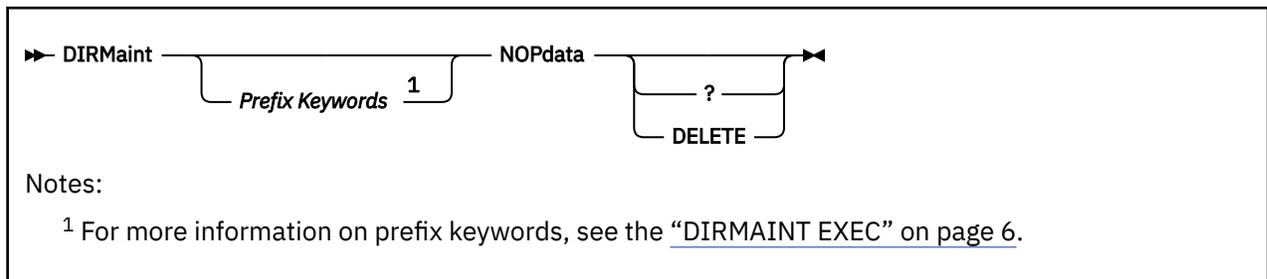
5. With Directory Network Authorization (DNA), a system administrator can configure and consolidate a virtual NIC device and its network properties in a secure, centralized location in z/VM's user directory.

The following NICDEF operands support DNA:

```
PORTNUMBER portnum  
PORTTYPE ACCESS | TRUNK  
VLAN vidset  
PROMISCUOUS | NO PROMISCUOUS  
PQUPLINKTX priority
```

By default, Directory Network Authorization (DNA) is enabled for use on z/VM but can be turned off dynamically by the system administrator with the SET VMLAN DNA CP command, or during IPL with the VMLAN DNA system configuration statement. When DNA is disabled, all DNA operands are ignored on the NICDEF statement. For more information on DNA, see [z/VM: Connectivity](#).

## NOPDATA



### Purpose

Use the NOPDATA operand of the DIRMAINT command to allow a user to add, delete, or query a NOPDATA statement in a directory entry.

### Operands

**?**

queries the directory to see if a NOPDATA statement currently exists within it.

### DELETE

specifies a request to delete the NOPDATA control statement from your directory entry.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa, MENU) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. For more information on the NOPDATA directory statement, see *z/VM: CP Planning and Administration*.
4. Requests to delete a nonexistent NOPDATA statement are rejected with a message.
5. Requests to add a NOPDATA statement are rejected with a message if the statement already exists.
6. If DELETE and ? are not specified, a NOPDATA control statement is added to the directory entry.

## NOTAPE



### Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the NOTAPE operand of the DIRMAINT command to cancel a tape mount request issued by the backup command (BACKUP TAPE operand).

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. If your installation is using backup tapes, the BACKUP TAPE processing will issue a message to your system operator requesting that one of the tapes be mounted on a drive, and attached to the DIRMAINT virtual machine as its 181.

Thereafter, DirMaint checks for the attachment of 181 approximately every 10 minutes. If the tape is still not attached after 10 minutes, appropriate messages are issued to the operator indicating the attach has not been satisfied.

This behavior can be modified through the DVHXTAPE EXEC for interfacing with your installations tape management program.

3. This operand can be issued if the tape cannot be mounted for any reason. It cancels the mount request and the tape portion of the backup processing.

## OFFLINE



### Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

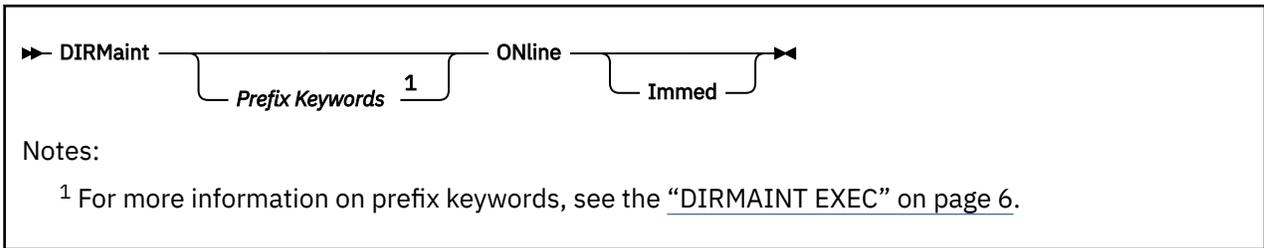
### Purpose

Use the OFFLINE operand of the DIRMAINT command to temporarily suspend placing the updated source directory online.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. This operand inhibits placing the updated source directory online. Normally, the source directory is placed online as scheduled in the CONFIG\* DATADVH file. The OFFLINE operand can be used to temporarily override this. This operand also inhibits CP update-in-place processing, if applicable.
3. The suspension period will last until the ONLINE operand is issued. The suspension period spans system shutdowns and restarts.
4. Manual invocation by the DIRECT operand is not inhibited in the OFFLINE state.

## ONLINE



### Purpose

Use the ONLINE operand of the DIRMAINT command to cancel a previous OFFLINE operand, allowing the updated source directory to be placed online.

### Operands

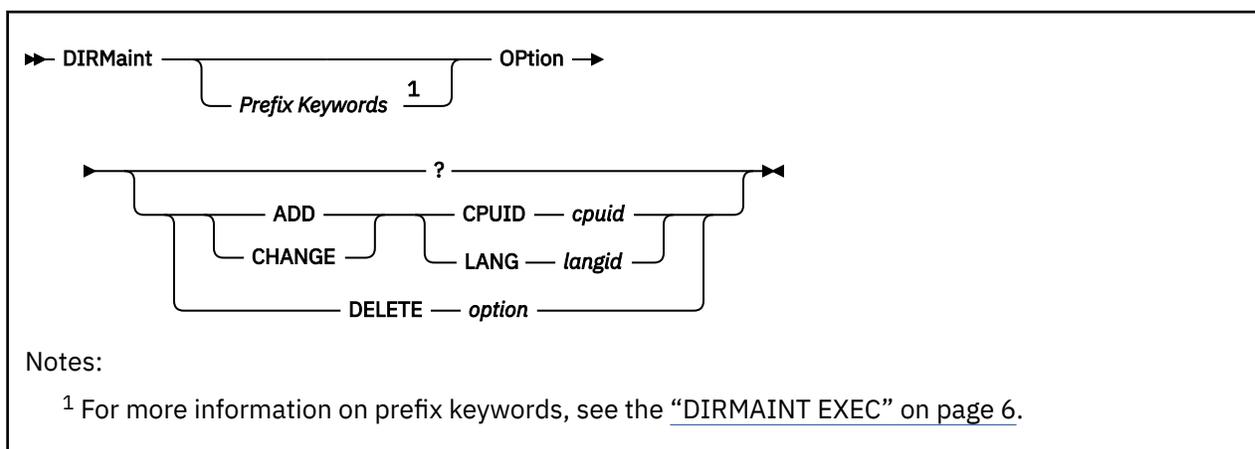
#### Immed

specifies that the source directory is to be placed online immediately.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. This operand cancels the effect of a previous OFFLINE operand, which prevented the placement of the updated source directory online.
4. This command causes a control file (OFFLINE CONTROL) to be deleted from the primary (and secondary if applicable) directory disk.
5. The configuration file (CONFIG\* DATADVH) is consulted to determine appropriate method of placing directory changes online.
6. If you want the source directory placed online immediately, include the IMMED option with the operand. Otherwise, the directory will be placed online as scheduled in the CONFIG\* DATADVH file, or by explicitly issuing the DIRECT operand.

## OPTION



### Purpose

Use the OPTION operand of the DIRMAINT command to enable you to request that certain CP options be added to, changed on, or deleted from the OPTION control statement in a profile, user, identity, or subconfiguration entry. Once the requested options are in the directory control statement, they will be in effect for each subsequent logon. The OPTION operand also allows you to query the OPTION control statement in the directory entry.

**Note:** Some options can result in a significant increase in overhead. An installation normally exercises restraint in granting use of these options to the general user.

### Operands

?

requests that the current OPTION statement be returned in a message.

#### ADD

specifies that the option that follows is to be added to the OPTION control statement in the directory entry.

#### CHANGE

specifies that the option that follows is to be changed on the OPTION control statement in the directory entry.

#### DELETE

specifies that the option that follows is to be deleted from the OPTION control statement in the directory entry.

#### *option*

specifies one of these options:

##### **ACct**

specifies that the virtual machine can issue DIAGNOSE code X'4C' to generate accounting records.

##### **AFfinity**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

##### **APplmon**

specifies that the virtual machine can issue DIAGNOSE code X'DC'.

##### **BMX**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

## OPTION

### **CFUSER**

specifies the user is authorized to connect to a Coupling Facility service machine.

### **CFVM**

specifies the user is a Coupling Facility service machine. CFVM may not be specified with RMCHINFO.

### **CHPIDVirtualization OFF**

CHPID virtualization is not active for the virtual machine. This is the default if CHPIDVirtualization is not included on either the GLOBALOPTS statement or the user's OPTION statement.

### **CHPIDVirtualization ONE**

Single path CHPID virtualization is active for the virtual machine.

### **COMsrv**

authorizes the indicated virtual machine to act as a communication server.

### **CONceal**

places the user in a protected application at logon time.

### **CPuid *cpuid***

provides a processor identification (CPUID) in hexadecimal to be stored in response to the STIDP instruction. The *cpuid* value is required for an ADD or CHANGE, but is prohibited for a DELETE.

### **CRYMeasure**

authorizes the specified virtual machine to obtain crypto measurement data from the crypto hardware on the real machine. Note that the RMchinfo option should **not** be specified when CRYMeasure is specified.

The CRYMeasure operand is not allowed in a subconfiguration entry.

### **DEVInfo**

specifies that the virtual machine can issue DIAGNOSE code X'E4', subcodes X'00' and X'01'.

### **DEVMaint**

specifies that the virtual machine can issue DIAGNOSE code X'E4', subcodes X'02' and X'03'.

### **DIAG200**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### **DIAG88**

specifies that the virtual machine can issue DIAGNOSE code X'88'.

### **DIAG98**

specifies that the virtual machine can issue DIAGNOSE code X'98'.

### **D84NOPAS**

specifies that the virtual machine can issue all subfunctions of DIAGNOSE code X'84' except LOGPASS and MDISK, without verification of the existing logon password.

### **Ecmode**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### **IGNMAXU**

indicates that this virtual machine can log on to the system even if the number of users already logged on is equal to or greater than the maximum allowed.

### **Isam**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### **LANG *langid***

identifies the language that should be set for the virtual machine during logon. The *langid* value is required for an ADD or CHANGE, but is prohibited for a DELETE.

### **LINKMSG**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**LKFAC**

indicates that the specified virtual machine is authorized to use the multipath lock facility RPQ simulation support.

**LNKExclu**

specifies that the virtual machine is authorized to use the stable and exclusive access modes.

**LNKNopas**

specifies that the virtual machine is authorized to link to any other virtual machine's DASD without password authorization.

**LNKStabl**

specifies that the virtual machine is authorized to use the stable access modes.

**LXAPP**

specifies the virtual machine can access the SE hard drive using DIAGNOSE X'2C4'. See [z/VM: CP Programming Services](#) for more information on DIAGNOSE X'2C4'.

**MAIntccw**

authorizes the specified virtual machine to use diagnostic CCWs.

**MAXConn *maxno***

specifies the maximum number of IUCV and APPC/VM connections allowed for this virtual machine. The *maxno* value is required for an ADD or CHANGE, but is prohibited for a DELETE.

**MAXVmcfi *maxno***

specifies the maximum number of VMCF inbound messages, including those initiated by SMSG, plus IDENTIFY final response interrupts that may be queued for processing on this virtual machine. The *maxno* value is required for an ADD or CHANGE, but is prohibited for a DELETE.

**MIh**

specifies that CP simulate an interrupt for the virtual machine whenever it detects a missing interrupt condition for an I/O operation.

**NETAccounting**

specifies that Network Data Transmission account records (type 0C) should be generated for this user.

**NETRouter**

specifies that this user is a network router, and Network Data Transmission account records (type 0C) should be generated for this user, and any user that communicates with this user over a virtual LAN.

**NOChain**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**NOMdcfs**

specifies that the virtual machine can use minidisk cache at a rate that is not limited by the fair share limit.

**NOMEMAssist**

Specifies that MEMASSIST support is disabled at logon time. Without this option, MEMASSIST support is enabled by default at logon time if the facility is available on the system.

**NOVF**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**Quickdsp**

causes a virtual machine to be added to the dispatch list immediately when it has work to do.

**REaltimer**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**RMchinfo**

indicates that the specified virtual machine is authorized to access real-machine configuration information, without regard to the virtual machine's configuration. It should be used when OSA/SF

## OPTION

(or any other application that may require real machine configuration information) is running under CMS. RMCHINFO should not be specified for guest operating systems.

The RMCHINFO operand is not allowed in subconfiguration entries.

### **SEorig**

specifies that the virtual machine can issue DIAGNOSE X'F8', subfunction X'00'.

### **STFirst**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### **STGexempt**

specifies that the virtual machine is exempt from CP free storage limit detection.

### **STHYI-Util**

authorizes the virtual machine to invoke utilization related functions of the Store Hypervisor Information (STHYI) instruction.

This operand is not allowed in subconfiguration entries.

### **STHYI-Guest**

authorizes the virtual machine to invoke guest related functions of the Store Hypervisor Information (STHYI) instruction.

This operand is not allowed in subconfiguration entries.

### **STHYI-Respool**

authorizes the virtual machine to invoke resource pool related functions of the Store Hypervisor Information (STHYI) instruction.

This operand is not allowed in subconfiguration entries.

### **SVCaccl**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### **SVCoff**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### **SVC76VM**

specifies that errors are not recorded by CP.

### **SVMstat**

specifies that the virtual machine is a service virtual machine. The monitor data records associated with this virtual machine include the SVMSTAT setting. The only purpose is to allow products that process monitor data to report on service virtual machines separate from end-user virtual machines. No other operations, such as transaction or wait state classification, are affected by this operand.

The SVMstat operand is not allowed in subconfiguration entries.

### **Todenable**

specifies that the user may change the virtual machine's time-of-day clock with the SCK instruction.

### **VCunoshr**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### **VMsave**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### **XMEM**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

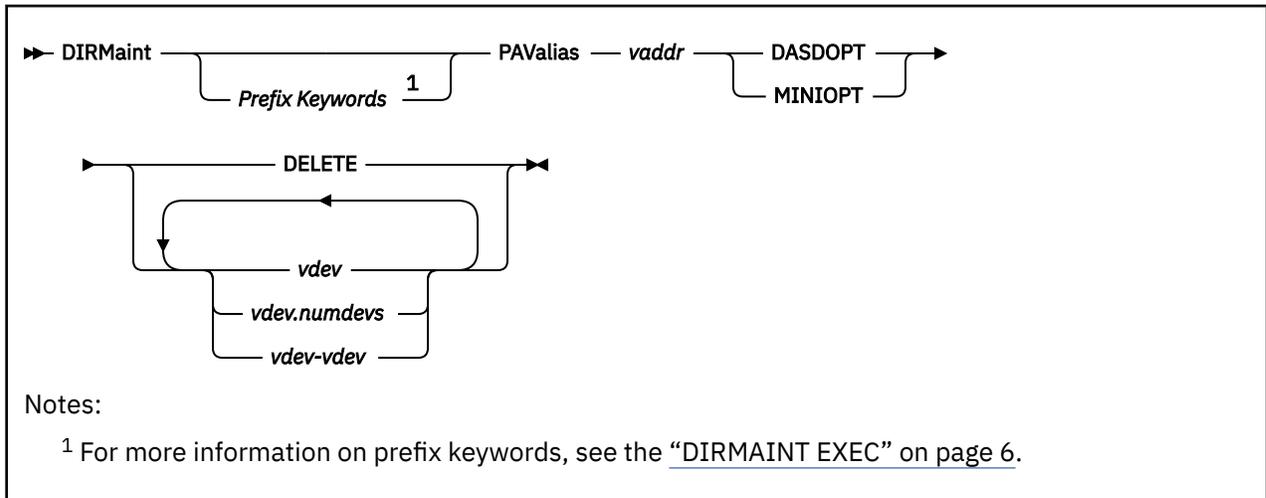
**370e**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. You may delete any existing values from the OPTION directory statement, however, you can only add or change those values that can be changed after logon using a CP SET, CMS SET, or other system commands. To add or change other values requires the use of the privileged SETOPTN command.
3. Not every option is valid for every release or in every type of directory entry. For more information on the OPTION directory statement, see [\*z/VM: CP Planning and Administration\*](#).

## PAVALIAS



### Purpose

Use the PAVALIAS operand of the DIRMAINT command to request that the PAVALIAS operand of the DASDOPT or MINIOPT directory statement associated with a specific device be added, changed, or deleted in a user, identity, or subconfiguration entry.

### Operands

#### **vaddr**

is the virtual address of the device with which the DASDOPT or MINIOPT directory statement is associated.

#### **DASDOPT**

specifies the virtual address is associated with a DASDOPT statement.

#### **MINIOPT**

specifies the virtual address is associated with a MINIOPT statement.

#### **DELETE**

specifies the PAVALIAS operand should be deleted from the DASDOPT or MINIOPT statement.

#### **vdev**

#### **vdev.numdevs**

#### **vdev-vdev**

specifies a single virtual device (*vdev*), a virtual device combined with a decimal range count (*vdev.numdevs*), or a range of virtual devices (*vdev-vdev*) to be added or updated on the DASDOPT or MINIOPT statement.

#### **Note:**

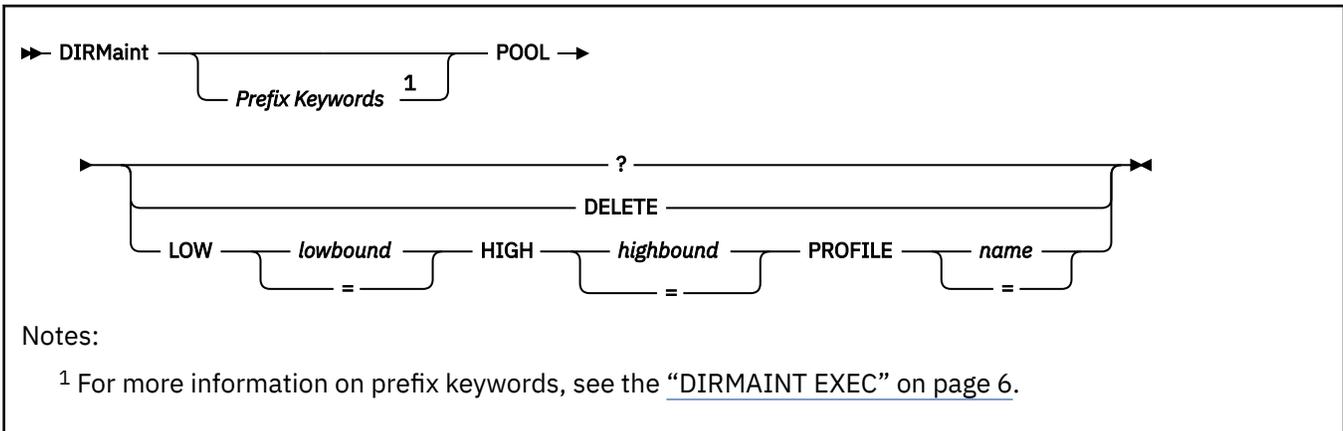
1. The device numbers entered must be hexadecimal numbers between X'0000' and X'FFFF'.
2. *numdevs* must be a decimal number.
3. Multiple device specifications must not overlap.
4. If a PAVALIAS operand is already specified on the associated DASDOPT or MINIOPT statement, it will be replaced.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOSys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.

2. For more information on the DASDOPT and MINIOPT directory statements, see [z/VM: CP Planning and Administration](#).

## POOL



Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC” on page 6](#).

### Purpose

Use the POOL operand of the DIRMAINT command to set or alter a POOL statement following a USER or IDENTITY statement. The POOL statement allows a set of virtual machines to be defined with the same configuration or characteristics.

### Operands

?

requests that the current POOL statement be returned in a message.

#### DELETE

requests that the current POOL statement be deleted.

#### LOW

specifies that the number to follow will represent the lowest value to use when assigning user IDs.

#### *lowbound*

specifies a number from 0 to 99999 that will be used as the suffix number for the first virtual machine.

#### HIGH

specifies that the number to follow will represent the highest value to use when assigning user IDs.

#### *highbound*

specifies a number from *lowbound* to 99999 that will be used as the suffix number for the last virtual machine.

#### PROFILE

specifies that the following field will be used as the profile definition for each virtual machine in the pool.

#### *name*

a valid profile entry in the directory.

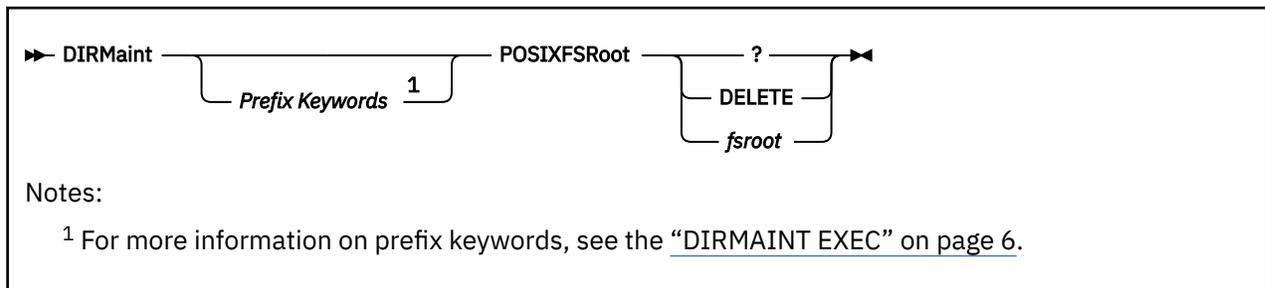
=

specifies that the existing value in the position should be maintained in the new directory statement.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the POOL directory statement, see [z/VM: CP Planning and Administration](#).
3. The target user ID of a POOL request must be 3 characters in length or less.

## POSIXFSROOT



### Purpose

Use the POSIXFSROOT operand of the DIRMAINT command to add, delete, alter, or query the FSROOT setting on a directory POSIXINFO statement in a profile, user, or identity entry.

### Operands

**?**

requests that the current settings of the FSROOT field on the POSIXINFO statement be returned in a message.

#### **DELETE**

requests that the FSROOT field of the POSIXINFO statement be deleted.

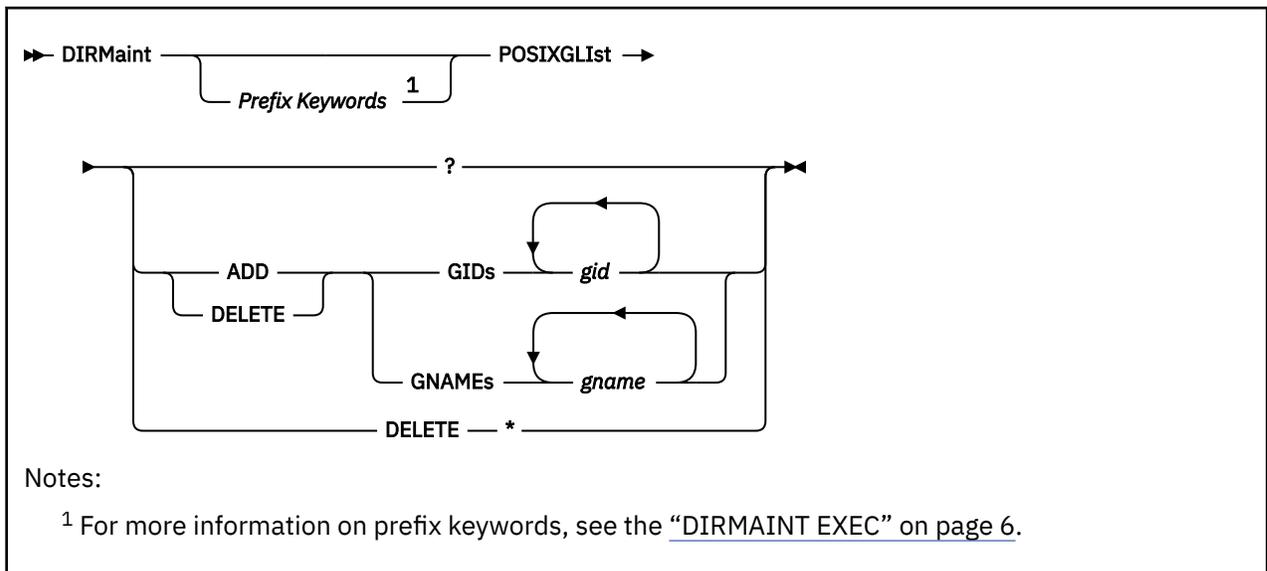
#### ***fsroot***

a string placed into the POSIXINFO statement in the FSROOT field.

### Usage Notes

1. You may use one or more of the prefix keywords (REQest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. To include imbedded blanks or special characters within a field requires the use of quoted strings.
3. For more information on specifying quoted strings and the POSIXINFO directory statement, see [z/VM: CP Planning and Administration](#).
4. If your system specifically has RACF installed as the ESM, see the [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

# POSIXGLIST



## Purpose

Use the POSIXGLIST operand of the DIRMAINT command to add, delete, or query POSIX groups on a POSIXGLIST directory statement in a profile, user, or identity entry. The POSIXGLIST directory statement lists POSIX groups of which a user is a member. Each group on the list can be specified by either group ID (*gid*) or group name (*gname*). POSIXGLIST statements are the primary source for the GIDs that form the user's supplementary group list.

## Operands

?

returns messages containing a list of the POSIX groups the user is currently a member of.

**ADD**

**DELETE**

either adds or deletes the specified POSIX groups.

**DELETE \***

requests that the entire POSIXGLIST statement be deleted.

**GIDs *gid***

specifies the POSIX group (by group ID) to be added or deleted in the range of 0 to 4294967295.

**GNAMEs *gname***

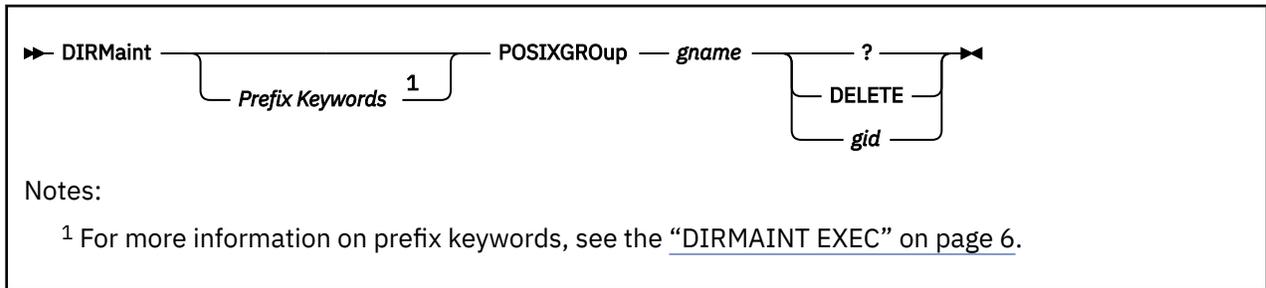
specifies the POSIX group (by group name) to be added or deleted.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the POSIXGLIST directory statement, see [z/VM: CP Planning and Administration](#).
3. If there are more than 31 GIDs and GNAMEs total on all of the POSIXGLIST statements in this directory entry, the order of them may be significant. For more information on controlling the order in which the GIDs and GNAMEs, see [“SUPGLIST”](#) on page 270.

4. If your system specifically has RACF installed as the ESM, see the [\*z/VM: Directory Maintenance Facility Tailoring and Administration Guide\*](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## POSIXGROUP



### Purpose

Use the POSIXGROUP command to add, delete, alter, or query a POSIXGROUP statement.

### Operands

#### *gname*

is the group name associated with this request.

?

requests that the current settings of the specified group be returned in a message.

#### DELETE

requests that the specified group be deleted.

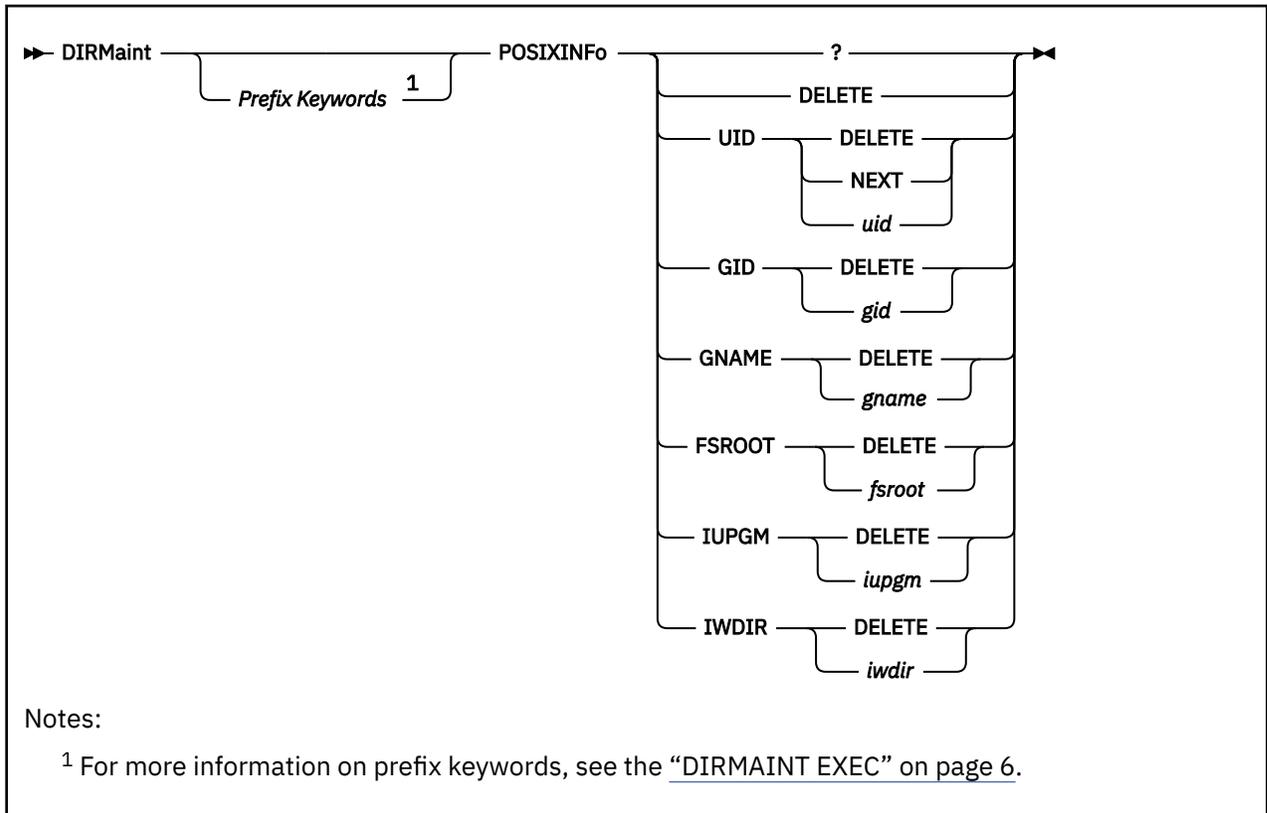
#### *gid*

a Group ID to be associated with the specified *gname*. If the group does not exist one is added. If it does, it is replaced with the new setting.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. These files will create the \$DIRGRP\$ DIRMPART file if none exists. Also, this file contains the GLOBALDEFS statement that is used to define the GLOBALOPTS and POSIXGROUP statements.
3. For more information on the POSIXGROUP directory statement, see [z/VM: CP Planning and Administration](#).
4. If your system specifically has RACF installed as the ESM, see the [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

# POSIXINFO



## Purpose

Use the POSIXINFO operand of the DIRMAINT command to add, delete, alter or query the settings on a directory POSIXINFO statement in a profile, user, or identity entry.

## Operands

**?**

requests that the current settings of your POSIXINFO statement be returned in a message.

### DELETE

specifies that the entire POSIXINFO statement be deleted or the listed field should be deleted. The action depends on the context in which the keyword is specified.

### UID

specifies that the POSIX user ID field of the POSIXINFO statement is to be set or reset.

### NEXT

requests that a UID be chosen from the POSIX\_UID\_AUTO\_RANGE setting in the configuration file.

### GID

specifies that the POSIX Group ID field of the POSIXINFO statement is to be set or reset.

### GNAME

specifies that the POSIX Group Name field of the POSIXINFO statement is to be set or reset.

### FSROOT

specifies that the File System Root field of the POSIXINFO statement is to be set or reset.

### IUPGM

specifies that the Initial User Program field of the POSIXINFO statement is to be set or reset.

**IWDIR**

specifies that the Initial Working Directory field of the POSIXINFO statement is to be set or reset.

**uid**

a value to be placed into the POSIXINFO statement in the UID field.

**gid**

a value to be placed into the POSIXINFO statement in the GID field.

**gname**

a value to be placed into the POSIXINFO statement in the GNAME field. The GNAME is listed on a POSIXGROUP statement that has occurred near the top of the directory (before any directory entry definitions) and resolves to a GID.

**fsroot**

a string to be placed into the POSIXINFO statement in the FSROOT field. Quoted strings may be used (see usage notes [“2”](#) on page 194 and [“3”](#) on page 194).

**iupgm**

a string to be placed into the POSIXINFO statement in the IUPGM field. Quoted strings may be used (see usage notes [“2”](#) on page 194 and [“3”](#) on page 194).

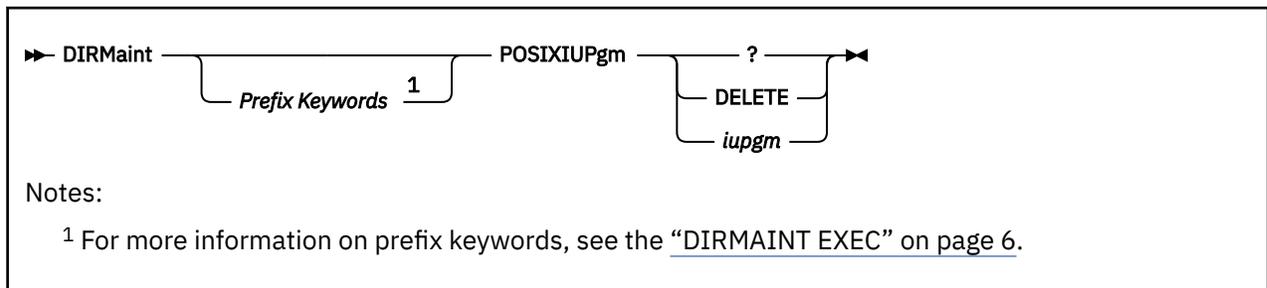
**iwdir**

a string to be placed into the POSIXINFO statement in the IWDIR field. Quoted strings may be used (see usage notes [“2”](#) on page 194 and [“3”](#) on page 194).

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. To include imbedded blanks or special characters within a field requires the use of quoted strings.
3. For more information on specifying quoted strings and the POSIXINFO directory statement, see [z/VM: CP Planning and Administration](#).
4. Since the GNAME resolves to a GID, these two fields are mutually exclusive. Setting one field clears the other.
5. If your system specifically has RACF installed as the ESM, see the [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## POSIXIUPGM



### Purpose

Use the POSIXIUPGM operand of the DIRMAINT command to add, delete, alter, or query the IUPGM (Initial User Program) setting on a directory POSIXINFO statement in a profile, user, or identity entry.

### Operands

**?**

requests that the current settings of your POSIXINFO IUPGM field be returned in a message.

#### **DELETE**

requests that the IUPGM field of the POSIXINFO statement be deleted.

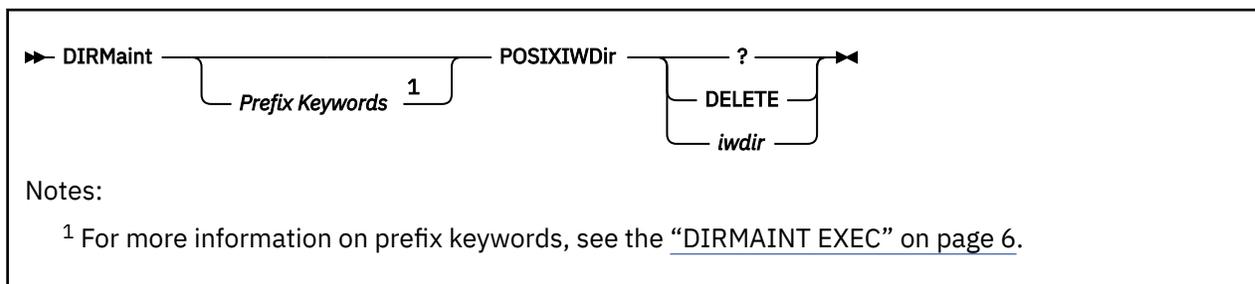
#### ***iupgm***

a string placed into the POSIXINFO statement in the IUPGM field.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. To include imbedded blanks or special characters within a field requires the use of quoted strings.
3. For more information on specifying quoted strings and the POSIXINFO directory statement, see [z/VM: CP Planning and Administration](#).
4. If your system specifically has RACF installed as the ESM, see the [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## POSIXIWDIR



### Purpose

Use the POSIXIWDIR operand of the DIRMAINT command to add, delete, alter, or query the IWDIR (Initial Working Directory) setting on a directory POSIXINFO statement in a profile, user, or identity entry.

### Operands

**?** requests that the current settings of your POSIXINFO IWDIR field be reported in a message.

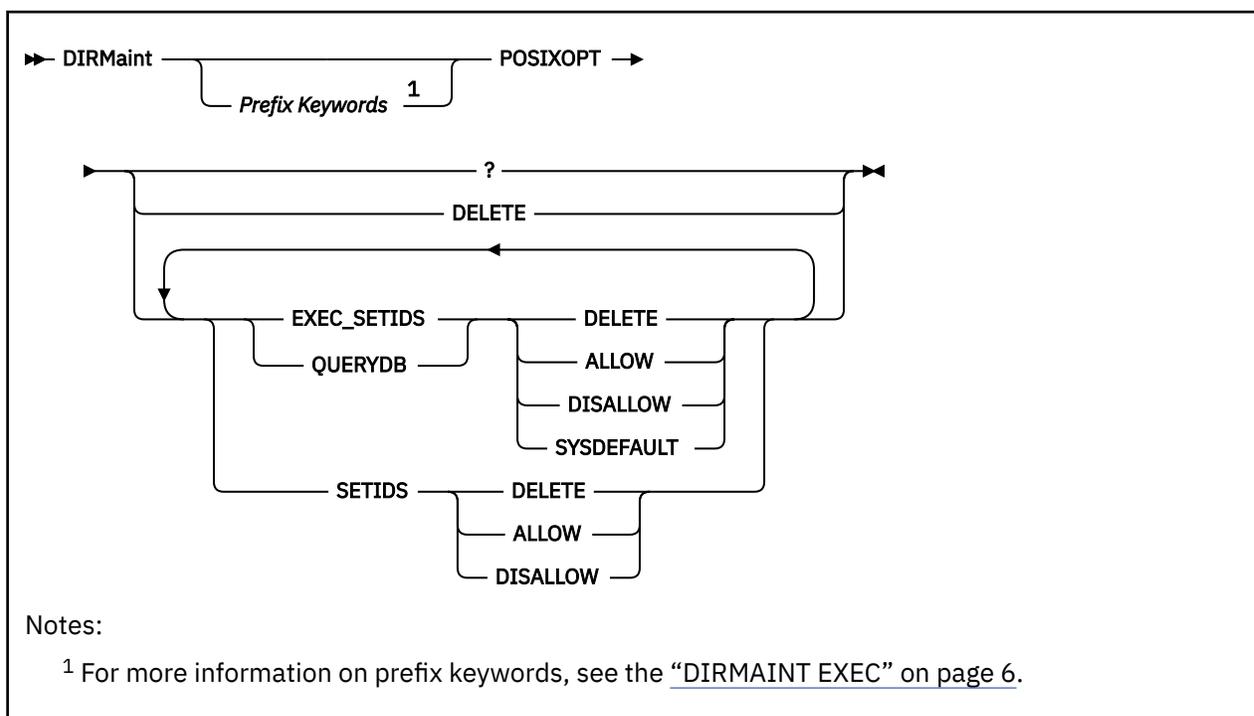
**DELETE** requests that the IWDIR field of the POSIXINFO statement be deleted.

***iwdir*** a string placed into the POSIXINFO statement in the IWDIR field.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. To include imbedded blanks or special characters within a field requires the use of quoted strings.
3. For more information on specifying quoted strings and the POSIXINFO directory statement, see [z/VM: CP Planning and Administration](#).
4. If your system specifically has RACF installed as the ESM, see the [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

# POSIXOPT



## Purpose

Use the POSIXOPT operand of the DIRMAINT command to add, delete, alter, or query a POSIXOPT statement in a profile, user, or identity entry.

## Operands

**?**  
requests that the current settings of the POSIXOPT statement be returned in a message.

**DELETE**  
when specified alone (without being used in conjunction with a specific field), this operand requests that the POSIXOPT statement be deleted. When specified in conjunction with a specific field, this operand requests that the specific field being mentioned be deleted.

**EXEC\_SETIDS**  
specifies whether the user is authorized to have their POSIX security values changed on behalf of a POSIX EXEC() call designating a file with the *setuid* or *setgids* attributes.

**QUERYDB**  
specifies whether the user is authorized to query another user's POSIX database information.

**SETIDS**  
specifies whether the user is authorized to set another user's POSIX UIDs or GIDs.

**ALLOW**

**DISALLOW**

specifies whether the preceding specified action is to be allowed or disallowed.

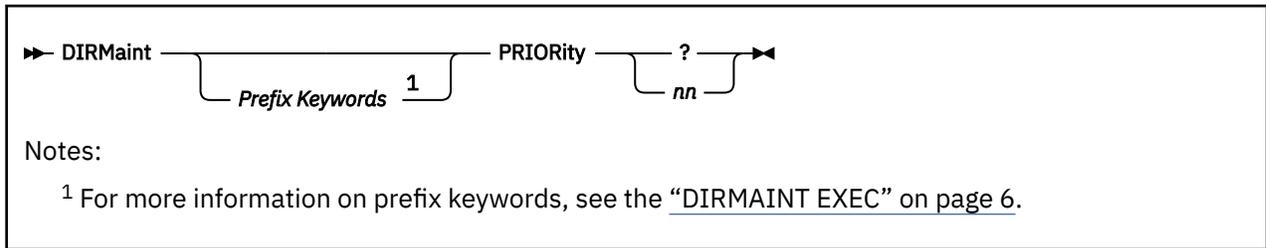
**SYSDEFAULT**

specifies the system default authorization.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the POSIXOPT directory statement, see [\*z/VM: CP Planning and Administration\*](#).
3. If your system specifically has RACF installed as the ESM, see the [\*z/VM: Directory Maintenance Facility Tailoring and Administration Guide\*](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

# PRIORITY



## Purpose

Use the PRIORITY operand of the DIRMAINT command to query or change the CP dispatching priority in a user or identity directory entry.

## Operands

**?**

requests that the current priority be displayed.

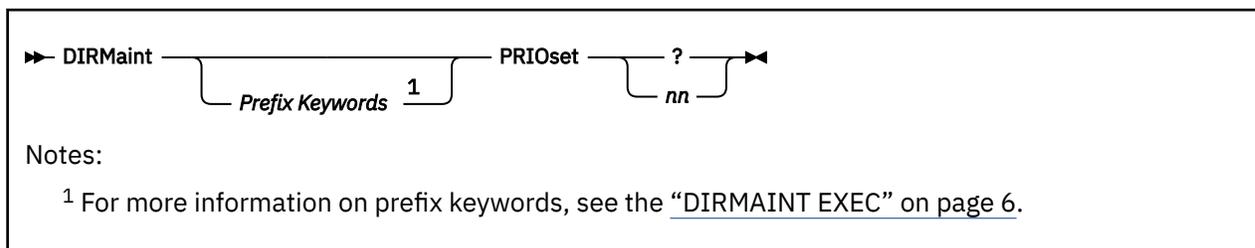
**nn**

is a two-digit decimal number that is to be the new CP dispatching priority. The highest priority that can be given a user is 01, the lowest (slowest) is 99. The default is 64.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOSys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. PRIOSET and PRIORITY are synonymous with SETPRiority. SETPRiority is the preferred operand. The others are retained for compatibility with previous releases.
3. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## PRIOSET



### Purpose

Use the PRIOSET operand of the DIRMAINT command to query or change the CP dispatching priority in a user or identity directory entry.

### Operands

**?**

requests that the current priority be displayed.

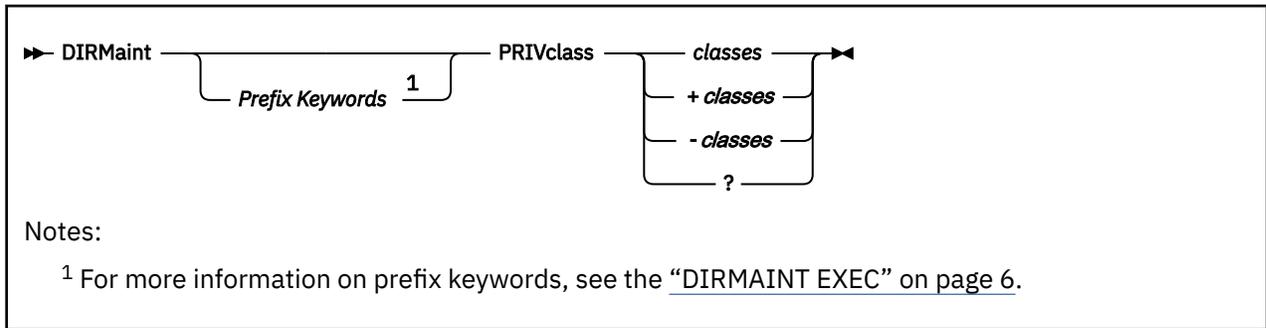
**nn**

is a two-digit decimal number that is to be the new CP dispatching priority. The highest priority that can be given a user is 01, the lowest (slowest) is 99. The default is 64.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. PRIOSET and PRIORITY are synonymous with SETPRiority. SETPRiority is the preferred operand. The others are retained for compatibility with previous releases.
3. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

# PRIVCLASS



## Purpose

Use the PRIVCLASS operand of the DIRMAINT command to assign, query, or delete CP privilege classes in a profile, user, or identity directory entry.

## Operands

### **classes**

specifies all the privilege classes which are being assigned to this directory entry. Any classes currently existing, including class G, will be replaced by those specified on this command.

### **+classes**

specifies the privilege classes to be added to this directory entry. These additional privilege classes will be added to those already existing in the directory entry.

### **-classes**

specifies the privilege classes to be deleted from this directory entry. Any other existing privilege classes will remain.

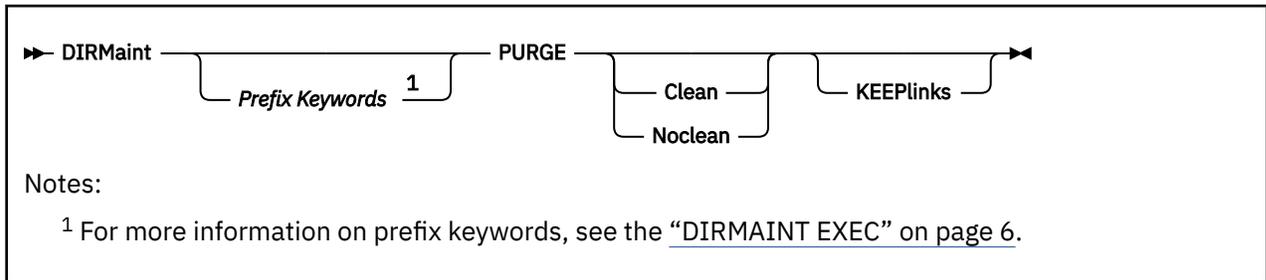
### **?**

specifies that a message should be returned listing the privilege classes currently assigned in the directory entry.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the CLASS directory statement and how it relates to the USER or IDENTITY statement, see [z/VM: CP Planning and Administration](#).
3. The classes supplied with the command must be specified as a contiguous string of letters and numbers.
4. When using the **+classes** option, if any of the supplied classes are already in effect, the transaction is rejected and the directory entry is not updated.
5. When using the **-classes** option, if any of the supplied classes are not currently in effect, the transaction is rejected and the directory entry is not updated.
6. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## PURGE



### Purpose

Use the PURGE operand of the DIRMAINT command to remove a directory entry from the source directory.

### Operands

#### Clean

specifies that the released minidisk extents are to be transferred to a DATAMOVE machine for formatting, even if the DISK\_CLEANUP= YES option is not in effect.

#### Noclean

specifies that the released minidisk extents are to be made immediately available for reassignment, even if the DISK\_CLEANUP= YES option is in effect. (See Usage Note [“5”](#) on page 203.)

#### KEEPlinks

Requests that all links be left as currently defined.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The PURGE request will be rejected in these situations. Attempts to delete:
  - A profile if any INCLUDE statements in the directory reference it. Delete the INCLUDE statements first.
  - The user ID acting as the DIRMAINT server.
  - A user ID that is defined as a DATAMOVE machine.
  - A user ID that is defined as a SATELLITE machine.
  - A user ID that is the same as the user ID issuing the command.
  - A user ID that is locked or a user ID that has device locks pending.
4. Directory entries containing MDISK statements need to be handled by the DASD subsystem asynchronously. There may be a short delay between the completion of the PURGE command and the directory entry being removed from the source directory. This is to allow the DASD subsystem to release the extents or format and release the extents.

During this period of time commands targeting the directory will be rejected and the user's password will be set to NOLOG. A CP QUERY command issued against the user ID being purged may show that the user ID is still present. The invoker of the PURGE command will be notified when DASD processing has completed.

If cleanup is the normal mode of operation, it can be bypassed by the NOCLEAN option. If cleanup is not in effect, it can be forced for a given transaction by including the CLEAN option.

5. When you purge a user ID with full-volume minidisks that deliberately overlay other users' minidisks, do not use the CLEAN option. The default action is NOCLEAN for overlapping minidisks.

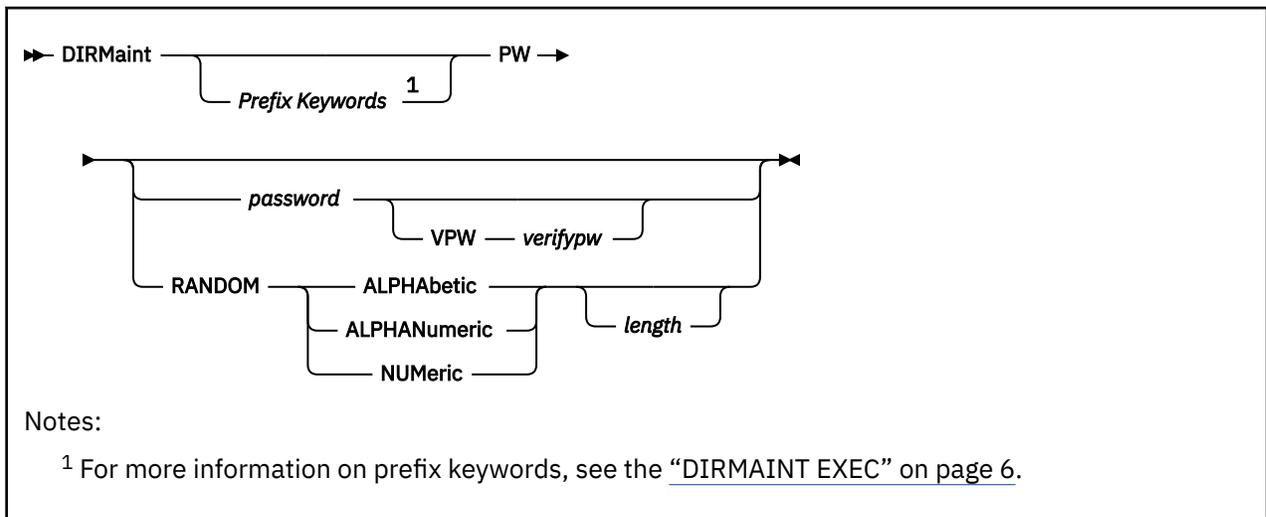
**Note:** If you use the CLEAN option, you will destroy those users' data.

6. Since profiles cannot contain MDISK statements, the NOCLEAN and CLEAN options are not valid when purging a profile and will be ignored.
7. PURGE processing removes any references to the user ID being purged from DirMaint control files. It also generates low priority work to remove any LINK statements that exist in other users' directory entries to minidisks owned by the user ID being deleted, unless the KEEPLINKS option is used. Purge processing also removes the subject user ID's DVHLINK FILE entries to other minidisks.

**Note:** Because the removal of LINK statements from other users' entries is scheduled as low priority work, it is possible that one of the users with a link statement could execute a LINK operand with a DELETE option before the low priority command executes. Another possibility is that the DLINK is unable to locate any LINK statements to be deleted. In either case, the low priority command fails since the link has already been removed. This *error* condition can be safely ignored.

8. When the PURGE operand is invoked with the CLEAN option all minidisks owned by the user are normally transferred to an available DATAMOVE virtual machine for formatting.
9. DEVNO MDISKS will not be formatted by DATAMOVE. If there is residual data that must be protected, it is the responsibility of the owning user to FORMAT the disk or otherwise obliterate the data before requesting the administrator to purge the user ID.
10. When a PURGE CLEAN is issued, any MDISK statements existing in the user ID's directory will be eliminated asynchronously through the DASD subsystem. This is done with an internally generated command called ZAPMDISK. For more information on ZAPMDISK, (Auxiliary DMDISK), see "Dasd Management" in the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*. If the ZAPMDISK fails, because someone else still has the link accessed (rc=104 - CP LINK error), the purge processing will not complete immediately. DATAMOVE will periodically retry the LINK and when it does succeed the ZAPMDISK will complete and the PURGE processing will continue.
11. If a DASD management work unit associated with this command becomes stalled or hung on the DATAMOVE machine, manual intervention may be necessary in order for the work unit to complete. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.
12. When a PURGE is issued against an identity entry, the identity entry and all related subconfiguration entries are purged.
13. When a PURGE is issued against a subconfiguration entry, the BUILD statement associated with it being purged is automatically removed from the related identity entry.
14. If the AT prefix is specified for a subconfiguration entry when the PURGE operand is issued, the AT keyword value will be used to select the DATAMOVE machine configured at the specified node. In this respect, the FORuser prefix keyword must specify the subconfiguration entry name in order to modify a subconfiguration entry with the PURGE operand.
15. If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## PW



## Purpose

Use the PW operand of the DIRMAINT command to change your CP logon password.

After entering the DIRMAINT PW command, you are normally prompted to enter your password twice.

## Operands

### *password*

specifies the value to be used as a password or passphrase. When entering the PW command password or passphrase on the VM command line, note that a passphrase containing embedded blanks must be surrounded by single quotes (with single quote characters within the passphrase itself doubled). When entering the password or passphrase on the PW command menu panel, up to 65 characters may be used. Longer passphrases must be specified by entering the DIRMAINT command on the VM command line or by issuing the DIRMAINT command in a z/VM Rexx EXEC. The password or passphrases will not appear in DirMaint maintained log files.



**Attention:** Use of this parameter may show up in console trace logs if tracing is turned on.

### **VPW** *verifypw*

specifies the same password again for typographical verification.

### **RANDOM**

requests that a random password be generated by calling the PASSWORD\_RANDOM\_GENERATOR\_USER\_EXIT.

### **ALPHAbetic**

requests that the random generator exit be provided with the keyword ALPHABETIC in the requested algorithm field.

### **ALPHANumeric**

requests that the random generator exit be provided with the keyword ALPHANUMERIC in the requested algorithm field.

### **NUMeric**

requests that the random generator exit be provided with the keyword NUMERIC in the requested algorithm field.

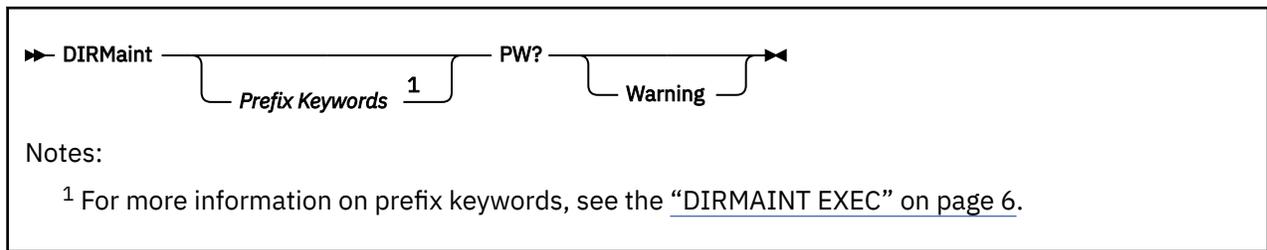
### *length*

requests that the random generator exit be provided with the value *length* as an optional parameter in the range 1 to 8.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, MENU, PROMPT) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. (Note that PROMPT is a default for this command, and will be assumed unless you override it with MENU.) The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. Your installation may have rules about password content, minimum length, or reuse of old passwords. These can be enforced by calling the PASSWORD\_SYNTAX\_CHECKING\_USER\_EXIT exit routine. For more information, see *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*.
4. Passwords generated by a call to the PASSWORD\_RANDOM\_GENERATOR\_USER\_EXIT exit are still passed through the PASSWORD\_SYNTAX\_CHECKING\_USER\_EXIT exit for verification.
5. Your installation may maintain a list of passwords whose use are restricted. The transaction will be rejected with an error message if your new password is on the restricted list.
6. All passwords are checked by the installations PASSWORD\_SYNTAX\_CHECKING\_EXIT, even though they have already passed the checking by the USER exit.
7. If the verification password does not match the first password value entered, the transaction is rejected.
8. Your installation may have a password change policy and have instituted password change monitoring and enforcement facilities. Password change enforcement can be automatic or manual, and you may receive messages pertaining to password change requirements. The contents of these messages depend upon which mode is in effect.
9. You can check how many days have elapsed since you last changed your password by inserting the command EXEC DIRMAINT PW? in your PROFILE EXEC. If you append the entry with the WARN parameter, you will only be notified when your password is near expiration.
10. Your installation is able to define the maximum number of days that may elapse between password changes. When you are within a few days of the limit, you may receive reminders that a change is due. Such notification may be by direct contact from the administrative staff, internal mail, or other means. Alternatively, the installation may elect to have the DIRMAINT virtual machine contact you directly.
11. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84') if it is enabled.
12. If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## PW?



### Purpose

Use the PW? operand of the DIRMAINT command to determine the date of the previous CP logon password change. Pending system password changes are also indicated in the response.

### Operands

#### Warning

specifies that a response is not returned to you unless your password is within the expiration warning period.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, MENU) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. On a system with an active External Security Manager (ESM) exit this command is rejected with a message since passwords are under control of the ESM.
4. The invocation of the PW? operand can be automated. The DIRMAINT command with the PW? operand can be placed in the PROFILE EXEC to automate the process as follows:

```
EXEC DIRMAINT PW?
```

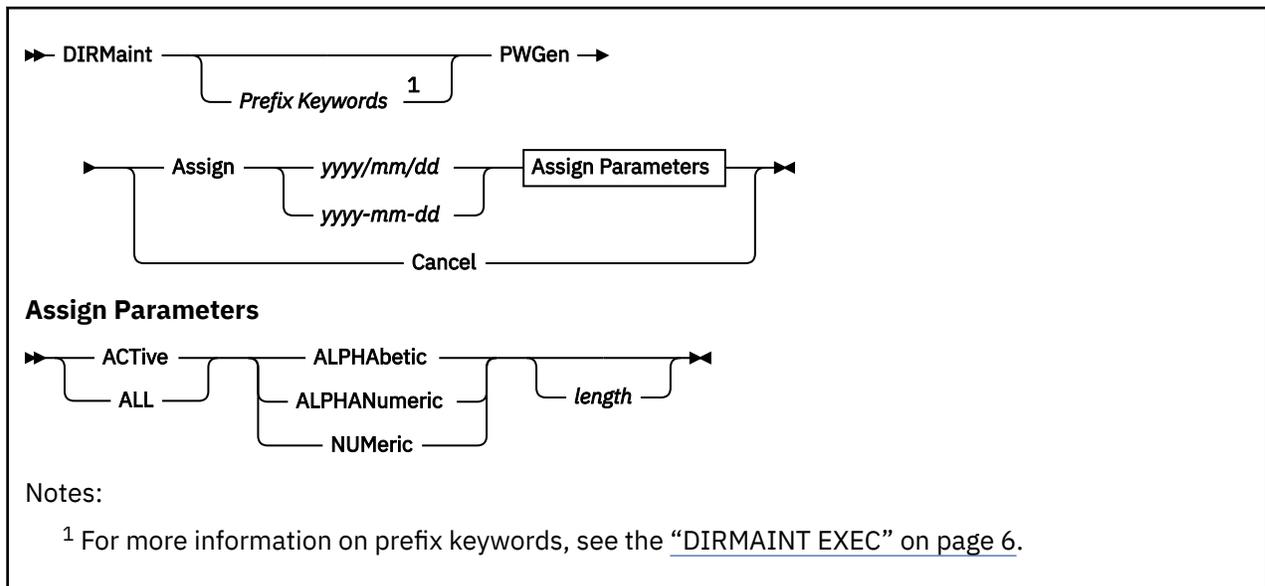
**Note:** The full spelling of DIRMAINT is required and it must be prefixed with EXEC.

If you only want a response when password expiration is near or a system password change is pending, add the WARNING option as follows:

```
EXEC DIRMAINT PW? WARNING
```

5. If the DIRM PW? command is issued while an administrator scheduled password change is pending, then the user will be required to enter their current password for validation. The response supplies both the new password and the effective date.  
If the DIRM PW? command is issued when no administrator scheduled change is pending, the user is not required to enter their current password. The response supplies the number of days remaining before the password must be changed by the user.
6. When the DIRMAINT PW? command is invoked, the user is informed (by a message) of the current age of their password and the number of days remaining until it is considered expired.
7. If a system password change event is pending, the user is also notified that the password will be changed. The date the change will become effective and the new password are also related in the message.

## PWGEN



### Purpose

Use the PWGEN operand of the DIRMAINT command to generate a set of random CP logon passwords with the specified attributes. These passwords are then associated with the user set as specified in the command invocation.

The new passwords remain pending until the specified date is reached. A pending change may be cancelled up to the effective date by specifying CANCEL as the first and only option.

### Operands

#### ASSIGN

causes DirMaint to generate a new password for users covered by the ALL or ACTIVE keywords.

#### *yyyy/mm/dd or yyyy-mm-dd*

specifies the date the new passwords go into effect. In order to allow time for users to be notified, this date should be 1 to 30 days in the future.

#### ACTive

indicates that new passwords will NOT be generated for any user now having a password of AUTOONLY, LBYONLY, NOLOG, or NOPASS.

#### ALL

indicates that new passwords will be generated for all users.

#### ALPHAbetic

indicates that new passwords will consist of alphabetic characters only.

#### ALPHANumeric

indicates that new passwords will consist of alphabetic and numeric characters.

#### NUMeric

indicates that new passwords will consist of numeric characters only.

#### *length*

specifies the length of the new passwords. The required length will depend on the implementation of your random password generation exit.

#### Cancel

causes DirMaint to cancel any pending PWGEN operation.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. Except in an emergency, it is recommended that the effective date be greater than 1 day in the future. This will allow time for users to be notified of the pending change. It also allows time to cancel the pending change if necessary.
3. Password generation requires extra processing. You may want to schedule use of the PWGEN operand for when the system is not heavily loaded. When the new passwords have been generated, a batch file called PASSWORD CHANGE is created. This file contains a list of user ID and password combinations that are due to be implemented.
4. When the passwords are created and before the passwords come into effect, the PASSWORD CHANGE file can be used for two purposes:
  - a. It can be used as input to an EXEC procedure you create to prepare notices to be sent to the affected user IDs.
  - b. It can be kept for reference and used to assist any user who was away during the password regeneration interval.

The file contains a comment as the first record (this record indicates the effective date). The remaining records are batch requests for each user to be acted on.

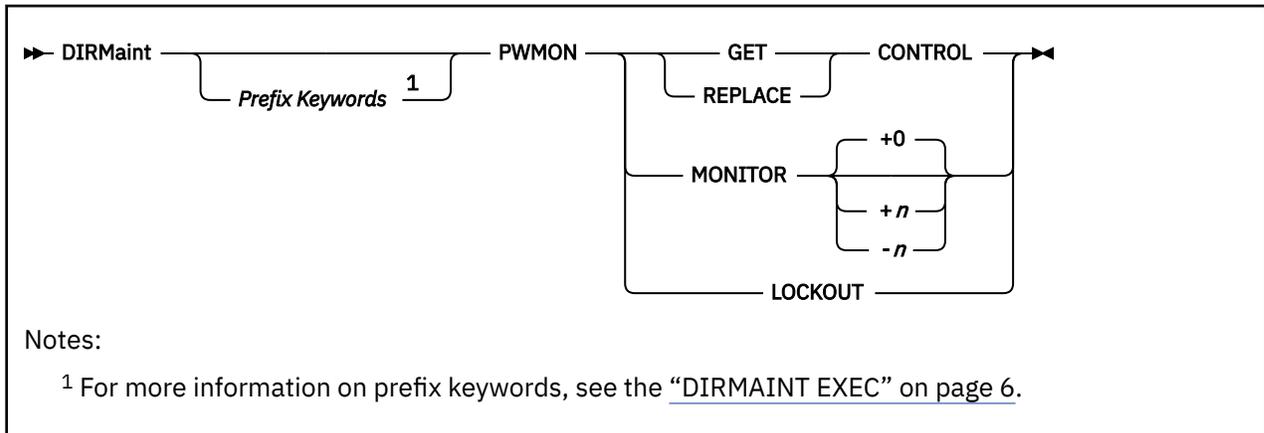
5. If the DIRM PW? command is issued while an administrator scheduled password change is pending, then the user will be required to enter their current password for validation. The response supplies both the new password and the effective date.
 

If the DIRM PW? command is issued when no administrator scheduled change is pending, the user is not required to enter their current password. The response supplies the number of days remaining before the password must be changed by the user.
6. Once the new passwords are pending, they become effective on the specified date at the scheduled backup time or at the next DirMaint IPL. When this procedure is being used, the backup time specified in the DIRMAINT DATADVH file should be as soon as possible after 00:00. This will ensure that the new passwords take effect very early on the specified date. Otherwise, any early logon users might attempt to log on before backup time and find that their old passwords are still in effect.
7. A pending PWGEN operation can be cancelled at any time between invocation and the effective date by issuing the command:

```
DIRM PWGEN CANCEL
```

If this is done, the administrator should notify all users that the expected password change has been cancelled.

## PWMON



### Purpose

Use the PWMON operand of the DIRMAINT command to control the CP logon password change date monitoring procedure.

### Operands

#### GET CONTROL

requests that the current copy of the PWMON CONTROL file be sent to your virtual card reader for review (and possible revision). The PWMON CONTROL file lists user IDs that are not responsible for changing their own logon passwords, such as the DIRMAINT virtual machine. Owners of these user IDs are also listed in this file (identified by their normal user ID) and notified when the password approaches its expiration date. This file also specifies whether the subject user ID is to be actually locked out, or is exempt from lockout with only warnings to be sent.

The PWMON CONTROL file is maintained by using an editor, all fields are delimited by one or more blanks. A general format of:

```
subject_userid YES alternate_userid alternate_node
```

or

```
subject_userid NO alternate_userid alternate_node
```

Where:

#### ***subject\_userid***

is the subject user ID, usually an unattended virtual machine.

#### **YES**

#### **NO**

YES specifies a lockout should occur, and NO specifies a lockout should not occur.

#### ***alternate\_userid***

is an alternate user ID that should receive warnings regarding the subject user ID.

#### ***alternate\_node***

is the node ID of the alternate user ID. If not specified, the local node is assumed.

The remainder of the line can be used for comments, such as authorization, date, and other appropriate information.

**REPLACE CONTROL**

requests that an original or modified PWWON CONTROL file, on one of your own accessed minidisks is to be installed in the DIRMAINT virtual machine and become effective immediately.

**MONITOR****MONITOR +n****MONITOR -n**

requests that the password change date monitoring procedure be executed. If MONITOR is specified alone, then date monitoring is performed with today's date as the reference point, this is the default +0. The +n (days) modifier in effect causes the reference point to be simulated as n days in the future, where as a -n modifier sets the reference point n days ago. For example, if it is Friday, then +3 simulates execution next Monday, and -4 simulates execution as it would have occurred last Monday. The overall operation of the password monitoring procedure is controlled by several statements in the CONFIG\* DATADVH files, while disposition of individual cases can be influenced by the PWWON CONTROL file previously discussed.

**LOCKOUT**

specifies that the PWWON LOCKLIST file that was returned to you by the monitoring procedure, and modified by you (deleting lines for users who should not be actually locked out), should be sent back to the DIRMAINT virtual machine and executed. This file consists of a series of SETPW *userid* NOLOG operands that will then be executed.

The PWWON LOCKLIST file is returned to you only when the LOCKOUT mode is manual. When LOCKOUT is specified as AUTOMATIC, then the operands in this file will be executed automatically after the monitoring procedure finishes.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords do not apply to this command. They are accepted without error, but otherwise ignored.
2. On a system with an active External Security Manager (ESM) exit this command is rejected with a message since passwords are under control of the ESM.
3. MONITOR +n or MONITOR -n should use the MANUAL LOCKOUT method.

```
PW LOCK MODE= MANUAL
```

```
PW WARN MODE= MANUAL
```

Otherwise the calculated cutoff date will be based upon the supplied +n or -n value.

## QLOG

---

**Notes:**

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

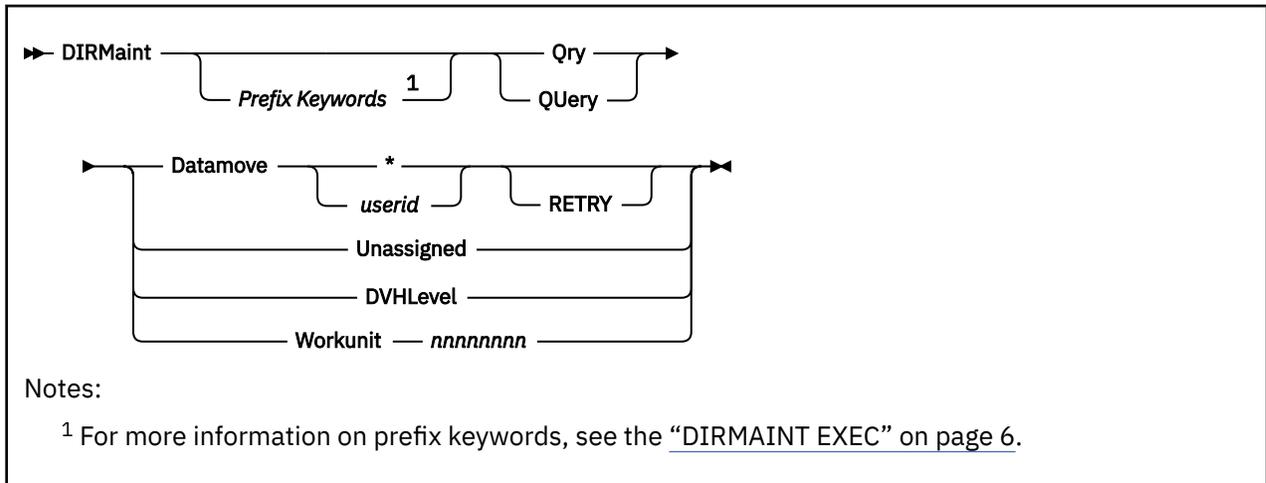
### Purpose

Use the QLog operand of the DIRMAINT command to retrieve any pending messages for the originating user.

### Usage Notes

You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.

## QUERY, QRY



### Purpose

Use the QUERY operand of the DIRMAINT command to request current system information from the DIRMAINT server.

### Operands

#### Datamove

requests that the current status of one or all DATAMOVE machines be reflected.

\*

requests that all DATAMOVE machines be queried.

#### userid

requests that a specific DATAMOVE machine be queried.

#### RETRY

requests that the current number of pending elements and the status of the specified DATAMOVE machine be displayed.

#### Unassigned

requests that the status of the current unassigned workunit queue be displayed.

#### DVHLevel

requests that the current DirMaint system level file be displayed.

#### Workunit nnnnnnnn

requests that detailed information associated with the specified Work Unit Control File be shown.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.

## REPLACE



### Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the REPLACE operand of the DIRMAINT command to place an updated version of a directory entry in the source directory. This operand is typically used after the entry has been extracted with the GET operand and updated in the requestor's virtual machine using an editor.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Prior to invoking this operand, you can use the GET operand to retrieve the directory entry. After loading the returned file, you can modify it as necessary with the CMS editor.
3. You cannot replace one directory entry type with a different directory entry type.
4. This GET, edit, and REPLACE cycle should not be used to add a new minidisk, or to change an existing one to a new size or extent. There are specific operands for those options. Use the AMDISK operand to add a new minidisk to a user ID, or the CMDISK/RMDISK operands to change an existing minidisk in terms of size, extent, and volume residence. You can directly change the minidisk access mode and access passwords using the MDISK operand.

While it is possible to add or change minidisks using the DIRMAINT REPLACE operand, there will be no extent validation, and automatic allocation is not performed. Also, all movement of data files between moved extents will have to be coordinated manually.

The only valid circumstance for using this method to change a minidisk is when it is to be expanded or contracted in place. For example, when a 10 cylinder minidisk is to be changed to one that is either 5 cylinders or 15 cylinders, but still is to begin at the same place on the same volume. When making this type of change, take every precaution to avoid loss of user files.

5. The DELETE option of the LINK operand is recommended when a LINK statement is to be deleted.
6. The GET, edit, and REPLACE cycle is, however, invaluable for modifying those statements in a directory entry that are not specifically supported by a DIRMAINT operand. Currently all statements are supported with the exception of comments. This method ensures that you will always have a means of performing directory changes involving any type of directory control statement.
7. The REPLACE operand automatically unlocks the directory entry, which has been locked since you issued the GET operand. While the entry was locked, users could not make any changes to it.
8. Any attempt to replace a directory entry that is not locked is rejected. This is done to minimize the possibility that this updated version would nullify an update made by the definition owner or another system administrator.
9. If the required syntax of the directory structure is violated the command will be rejected. For instance, if a USER statement were to appear inside a profile entry, the REPLACE command would be rejected.
10. If an INCLUDE statement in a replacement directory entry invokes a nonexistent profile, an error message will be issued and the entry will fail.

## REPLACE

11. If a BUILD statement is added for a nonexistent subconfiguration entry, the REPLACE will fail with an error message. Likewise, if a BUILD statement is removed for an existing subconfiguration entry, the REPLACE will fail with an error message.
12. When the updated definition of a user or identity entry contains STORAGE, MAXSTORAGE, or CLASS statements, their values will be moved to the corresponding fields in the USER or IDENTITY statement. For example, if after you retrieve and modify a copy of TEST it contains the following statements:

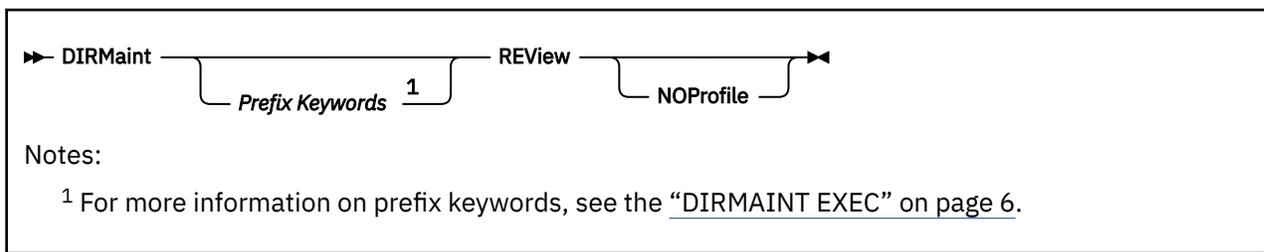
```
IDENT TEST A1B2C3
BUILD ON SYS1 USING SUBCONFIG TEST1
BUILD ON SYS2 USING SUBCONFIG TEST2
CLASS ABCDEFG
STORAGE 64M
MAXSTORAGE 256M
```

then when you replace the existing identity directory entry with this updated definition and retrieve a copy of the entry again, it will contain the following:

```
IDENT TEST A1B2C3 64M 256M ABCDEFG
BUILD ON SYS1 USING SUBCONFIG TEST1
BUILD ON SYS2 USING SUBCONFIG TEST2
```

13. The REPLACE operand causes the generation of a new checksum which is then filed with the entry.
14. If an existing minidisk is omitted from a directory entry that is replaced, or if the allocation information for an existing minidisk is changed in a directory entry being replaced, and there is no other minidisk in the directory that occupies the same extent as that occupied by the old minidisk in the replaced entry, DirMaint will consider that space as available for allocation to another minidisk (through either ADD, AMDISK, CMDISK or RMDISK).  
**Note:** This has not been formatted, even if DISK\_CLEANUP= YES is specified in the CONFIG\* DATADVH file.
15. Duplicate comment records starting with the string \*DVHOPT are restricted. Any such lines will be removed from the supplied user/profile entry. \*DVHOPT records are maintained by DirMaint. On ADD and REPLACE commands, only the first \*DVHOPT record is maintained in the directory entry.
16. If a DASD management work unit associated with this command becomes stalled or hung on the DATAMOVE machine, manual intervention may be necessary in order for the work unit to complete. See the [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#) for more information on how to diagnose and process work units which are in a stalled or retry state.
17. If your system specifically has RACF installed as the ESM, see the [z/VM: Directory Maintenance Facility Tailoring and Administration Guide](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## REVIEW



### Purpose

Use the REVIEW operand of the DIRMAINT command to allow you to review the control statements in your own directory entry. This allows you to check your virtual machine configuration and options, but not your passwords.

### Operands

#### NOProfile

specifies that if your directory entry includes a profile, you do not want to see it expanded.

Usually the contents of the profile are included inline, set off by these lines:

```
DVHRXV3355I The following records are included from profile: name
```

```
DVHRXV3355I The preceding records are included from profile: name
```

A profile is a group of control statements that are common to several directory entries. One INCLUDE statement invokes an entire profile, just as if every control statement in the profile were coded in your directory entry. Statements in your directory entry override statements in the profile.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, MENU, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The logon password for the subject user ID and the password field on all APPCPASS statements are always masked with X's in the REVIEW output. In general, all minidisk access passwords will also be masked in the same fashion. Minidisks with access passwords of ALL will be left intact in the REVIEW output. This is done to highlight that there is some type of universal access to these minidisks.
4. The list of all links to minidisks owned by the subject user ID may not be complete if the installation has identified one or more of the minidisks as candidates to be excluded. The data that is referenced when generating this section of the output comes from a file other than the source directory. The content of this file is determined by the installation. If known links appear to be missing from the REVIEW output, the system administrator should be contacted. It is likely that the minidisk in question is the subject of an entry in the DVHLINK EXCLUDE file. For more information, see [“USEROPTN”](#) on page 287.

## RLDCODE



Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the RLDCODE operand of the DIRMAINT command to cause the DIRMAINT virtual machine to reload its resident operating procedures. The processing invoked by the RLDCODE operand is also invoked automatically as an integral part of the system initialization by means of IPL.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The DIRMAINT virtual machine runs with most of its performance critical parts preloaded into virtual storage using either EXECLOAD (for EXECs, REXX pipeline stages, and XEDIT macros) or NUCXLOAD (for MODULE files).
3. If the FILE operand is used to replace a preloaded part, the FILE operation should be followed with this RLDCODE operand to put the revised version into operation. Otherwise, it will not be available for use until the next system restart.

## RLDDATA



### Notes:

- <sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

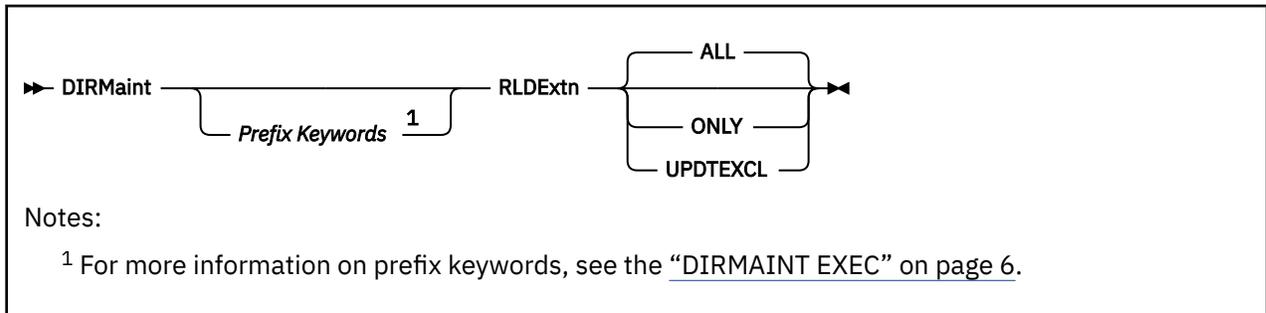
## Purpose

Use the RLDDATA operand of the DIRMAINT command to tell the DIRMAINT virtual machine to reload the data tables from the CONFIG\* DATADVH disk files. The processing invoked by the RLDDATA operand is also invoked automatically as an integral part of system initialization during IPL. The RLDDATA operand does not automatically invoke the RLDEXTN function.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The DIRMAINT virtual machine loads DIRMAINT's global variables at IPL time from the CONFIG\* DATADVH disk files. The CONFIG\* DATADVH disk files contain the basic installation options and parameters.
3. The RLDDATA operand will reload DIRMAINT's global variables from the CONFIG\* DATADVH file(s). Note that variables defined at IPL time which have been subsequently removed from the CONFIG\* DATADVH files will not be reset. To reset a variable to null, it must remain in the CONFIG\* DATADVH file with a null value.
4. The RLDDATA operand must not be used to reload a CONFIG\* DATADVH file with a DATAMOVE statement that contains a different user ID than that of the currently existing DATAMOVE virtual machine. To change the name of the DATAMOVE virtual machine:
  - a. Insure that the DATAMOVE machine you intend to delete is not busy. This can be accomplished by using the DIRM QUERY command or by looking at the DATAMOVE CONTROL file on the DirMaint 1DF disk.
  - b. Stop the DIRMAINT server.
  - c. Change the CONFIG\* DATADVH file
  - d. Initialize the DirMaint system with the DVHBEGIN command.
5. The EXTENT CONTROL file carries specifications for using minidisk volumes. Modifications to the EXTENT CONTROL file must be activated using the RLDEXTN operand.

## RLDXTN



### Purpose

Use the RLDXTN operand of the DIRMAINT command to reload the information in the EXTENT CONTROL file. It also provides a method for rebuilding the various volume control files on the system.

### Operands

#### ALL

All functions are performed. This is the default action.

The functions included are:

- Process the Extent Control file and form all the various internal use files from the extent control file.
- Rebuild the DASD volume control files from the source directory.
- Rebuild the link control file from the source directory.

#### ONLY

Only the extent control file is processed. This option does not rebuild the volume control and link files from the source directory. This option is appropriate when the EXCLUDE section of the Extent Control file was not altered. Placing any new excluded extents into the current volume control file will not be done if this option is used.

#### UPDTEXCL

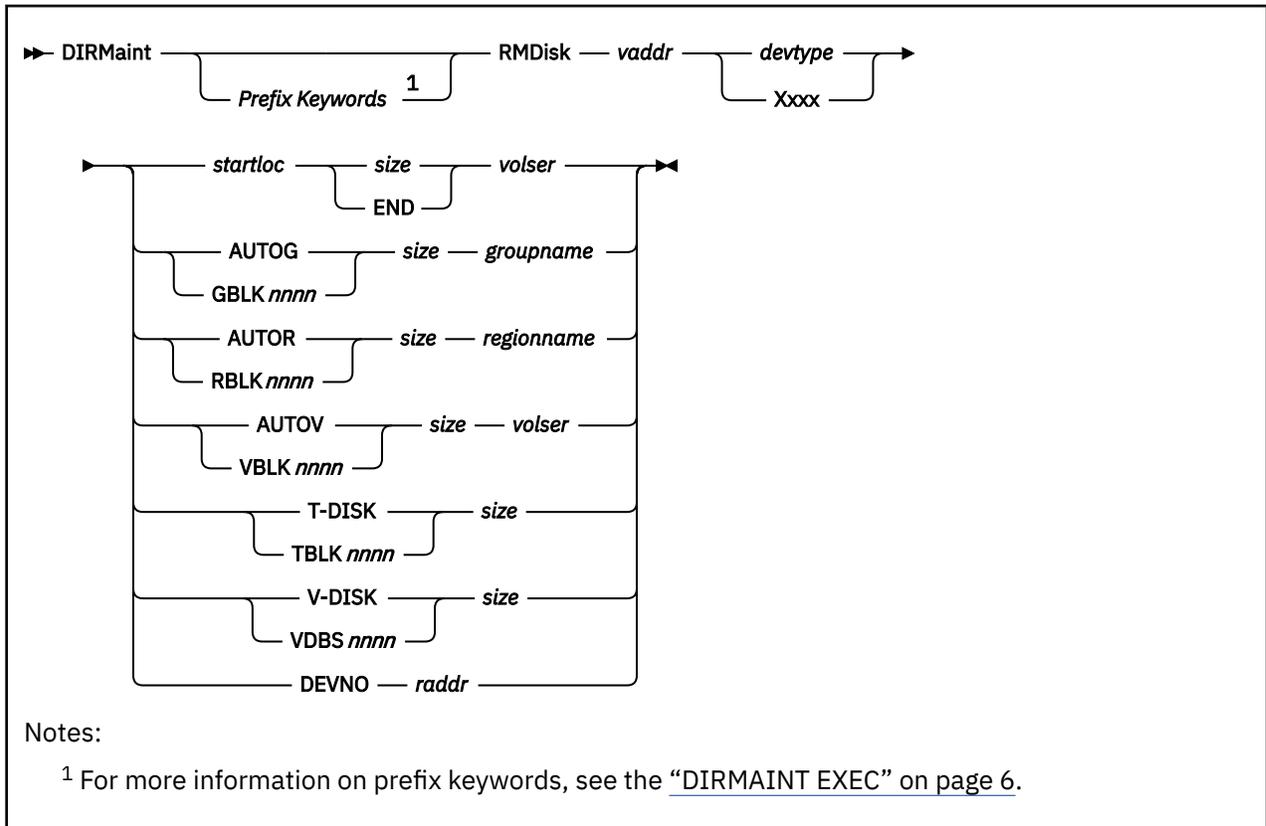
This option processes the extent control file and also manipulates the existing volume control files to reflect the new excluded extents. This function differs from ALL in that the source directory is not consulted. This may prove to be useful for very large installations.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The EXTENT CONTROL file carries specifications about how minidisk volumes are used. Its contents control the processing in the automatic minidisk allocation routines.
3. The RLDXTN operand can be used to activate an EXTENT CONTROL file that has been modified or to force a rebuild of your volume control file structures.
4. The UPDTEXCL option may not rebuild the control files from the source directory. For example, if your site allows administrators to directly manipulate the directory source without using DirMaint. The possibility exists that MDISK/LINK statements could be altered without the new values being reflected in DirMaint's internal files (the volume control files and the link file). Should this occur the volume control files should be rebuilt. Using the ALL option has the effect of synchronizing DirMaint's internal structure with the current directory and should be used if there is any possibility of direct source directory manipulation.

5. A DIRM RLDEXTN ONLY command is issued automatically during DVHBEGIN processing upon initialization or restart of the DIRMAINT service machine. If the DASD volume control files need to be rebuilt from the source directory, you must issue a DIRM RLDEXTN ALL command.

## RMDISK



### Purpose

Use the RMDISK operand of the DIRMAINT command to redefine an existing permanent minidisk in a user, identity, or subconfiguration entry.

### Operands

#### **vaddr**

is the virtual device address of the minidisk to be redefined.

#### **devtype**

is the device type of the volume to which the minidisk is assigned. Otherwise, if automatic allocation is used, specify the device type as *Xxxx*. The automatic allocation options include these operations:

- AUTOG and GBLK
- AUTOR and RBLK
- AUTOV and VBLK

The real device type will be taken from the REGIONS section in the EXTENT CONTROL file for that volser.

#### **startloc**

is a decimal cylinder relocation factor or a decimal FB-512 device block number that specifies the beginning of the minidisk extent corresponding to cylinder 0 or block 0 of the user's minidisk.

#### **AUTOG**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a volume within a specified group. The group must have been defined in the EXTENT CONTROL file, and the regions within the group must also be defined.

**GBLKnnnn**

is functionally equivalent to AUTOG except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: GBLK512, GBLK0512, GBLK1024, GBLK2048, GBLK4096, GBLK1K, GBLK2K, and GBLK4K. In addition, GBLK800 or GBLK0800 may be specified for device types 3350, 3375, and 3380.

The disk will not be formatted for CMS use by DirMaint unless the BLKSIZE or LABEL operands are used, but the number of CMS blocks of a given size per cylinder is used to calculate the number of CKD cylinders or FB-512 blocks necessary to satisfy the request.

**AUTOR**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a specified region or volume. The region must have been defined in the EXTENT CONTROL file.

**RBLKnnnn**

is functionally equivalent to AUTOR except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: RBLK512, RBLK0512, RBLK1024, RBLK2048, RBLK4096, RBLK1K, RBLK2K, and RBLK4K. In addition, RBLK800 or RBLK0800 may be specified for device types 3350, 3375, and 3380.

**AUTOV**

specifies that a starting CKD cylinder or FBA block number is to be automatically assigned at a suitable place on a specified volume. The volume must have been defined in the EXTENT CONTROL file.

**VBLKnnnn**

is functionally equivalent to AUTOV except that the size is specified in units of CMS 512, 800, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: VBLK512, VBLK0512, VBLK1024, VBLK2048, VBLK4096, VBLK1K, VBLK2K, and VBLK4K. In addition, VBLK800 or VBLK0800 may be specified for device types 3350, 3375, and 3380.

The disk will not be formatted for CMS use by DirMaint unless the BLKSIZE or LABEL operands are used, but the number of CMS blocks of a given size per cylinder is used to calculate the number of CKD cylinders or FB-512 blocks necessary to satisfy the request.

**T-DISK**

provides temporary disk space to a user upon logging on. This space is made available from preallocated system disk space. The owner must initialize or format this minidisk at each logon. It will be a part of the owner's virtual configuration until logoff or disk detachment, at which time the owner loses all files and the space is returned for reallocation to another user. If concerned about security, the owner should initialize or reformat the minidisk before releasing it.

**TBLKnnnn**

is functionally equivalent to T-DISK except that the size is specified in units of CMS 512, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: TBLK512, TBLK0512, TBLK1024, TBLK2048, TBLK4096, TBLK1K, TBLK2K, and TBLK4K.

**V-DISK**

provides virtual disk space to a user upon logging on. The V-DISK may already exist when the user links to it, providing other users that previously linked to it are still linked when the user logs on. Otherwise, the owner must initialize or format this minidisk at each logon. It remains a part of the owner's virtual configuration until logoff or disk detachment. When the last linked user logs off or detaches the V-DISK, the file space is returned for reallocation to another user.

**VDBSnnnn**

is functionally equivalent to V-DISK except that the size is specified in units of CMS 512, 1024, 2048, or 4096 blocks rather than in CKD cylinders or FBA blocks. Valid values are: VDBS512, VDBS0512, VDBS1024, VDBS2048, VDBS4096, VDBS1K, VDBS2K, and VDBS4K.

**size**

is a 1 to 8 digit decimal number specifying the size of the new minidisk in CKD cylinders or FBA blocks. For allocation requests involving block sizes, allocation will be converted to the exact number of FB-512 blocks for FBA devices or to the minimum number of cylinders needed to contain the specified number of CMS blocks for CKD devices.

**END**

specifies that the minidisk will extend from the assigned starting cylinder or block to the end (last cylinder or block) of the volume in which it is defined. This parameter is valid for fixed allocation only.

**groupname**

specifies the name of the DASD allocation group from which DirMaint will pick the *volser* and starting CKD cylinder or FBA block. Groups consist of one or more specific *regions*. This is valid for AUTOG and GBLKnnnn requests only. Groups must be defined in the EXTENT CONTROL file, and can be 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +.

**regionname**

specifies the name associated with the *region*, 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +, on which the minidisk is to be allocated. This is valid for AUTOG and RBLKnnnn requests. The named region must be defined in the REGIONS section of the EXTENT CONTROL file. The region defines a segment on a specific DASD volume for allocation.

**valid**

specifies the volume label of the DASD volume, 1 to 6 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +, on which the minidisk is to be allocated, or the reserved identifier of &SYSRES. This applies to fixed allocation (where a specific starting cylinder or block number is specified) and to automatic allocation on a volume (AUTOV, VBLKnnnn).

**Note:** The reserved identifier of &SYSRES should not be used as the *valid* in an SSI-enabled directory where the SSI option is used on the DIRECTORY statement. This will result in a DIRECTXA error and failure of the RMDISK command.

**DEVNO raddr**

specifies a full volume minidisk.

**raddr**

specifies the real device number of the full volume minidisk.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. It is not advisable to define a minidisk starting at real cylinder 0 unless it is to be used by OS ISAM. If it is to be used by OS ISAM, it must begin at real cylinder 0. If you do assign a minidisk beginning at real cylinder 0, the user who owns it must be aware that the minidisk label is the real label that the user and the system use to identify the disk. CP-owned volumes must not have minidisks beginning at real cylinder 0. Generally, CKD DASD extents should begin with cylinder 1 or higher, and FBA DASD extents should begin with block 32 or higher.
3. If there is no other minidisk in the directory that occupies the same extent as that occupied by the old minidisk in the replaced entry, DirMaint will consider that space as available for allocation to another minidisk (through either ADD, AMDISK, or CMDISK).

**Note:** This has not been formatted, even if DISK\_CLEANUP= YES is specified in the CONFIG\* DATADVH file.

4. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.
5. You may deliberately want to allocate a specific extent on a volume. If this extent would cross region boundaries or occupy a reserved area, you must issue the EXTNCHK OFF command before entering the RMDISK command.

6. The reserved identifier of `&SYSRES` should not be used as the *valid* in an SSI-enabled directory where the SSI option is used on the `DIRECTORY` statement. This will result in a `DIRECTXA` error and failure of the `RMDISK` command.
7. If the `AT` prefix is specified for a subconfiguration entry when the `RMDISK` operand is issued, the `AT` keyword value will be used to select the `DATAMOVE` machine configured at the specified node. In this respect, the `FORuser` prefix keyword must specify the subconfiguration entry name in order to modify a subconfiguration entry with the `RMDISK` operand.

### Examples

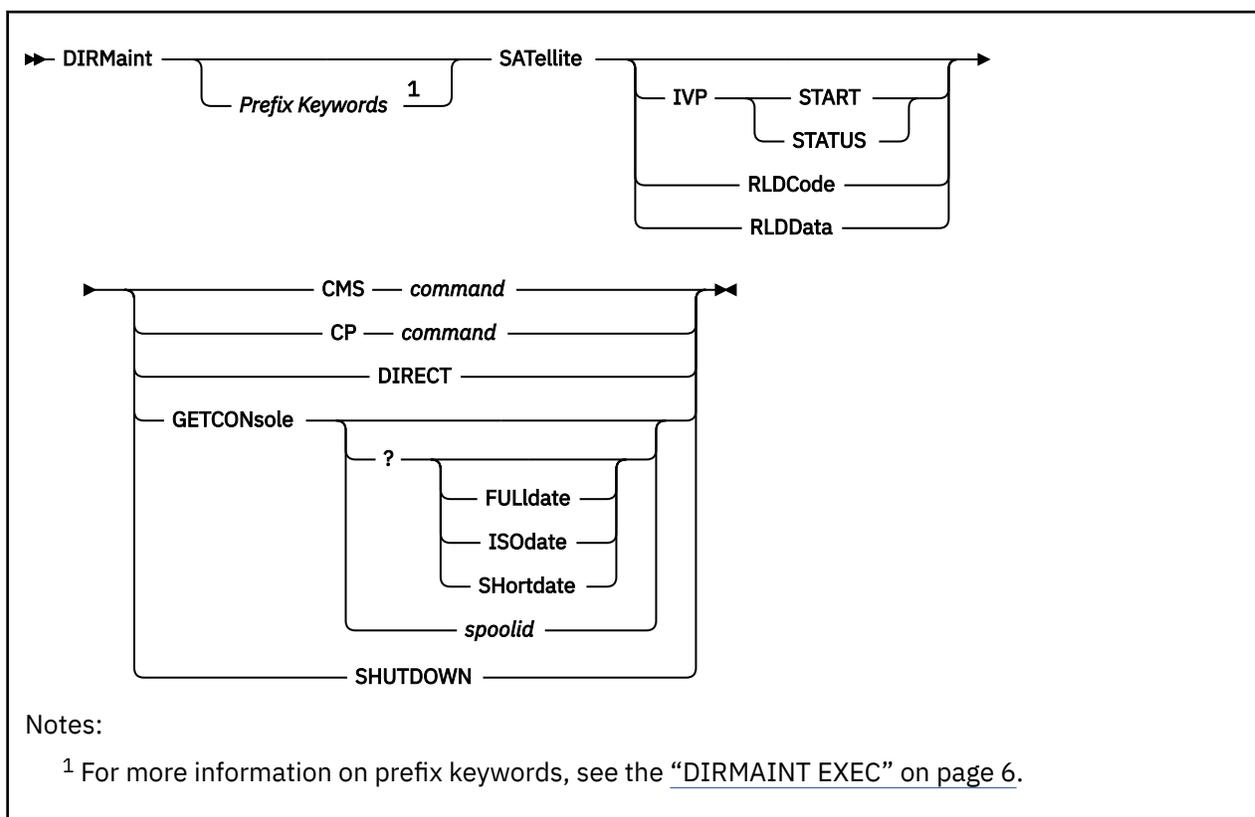
1. `DIRM FOR SMITH RMD 192 9345 438 6 VOL122`

The `DIRMAINT RMDISK` command redefines the 192 minidisk owned by SMITH (SMITH's 192 is redefined as a 9345 device starting at address 438 and extending for 6 cylinders on disk volume id VOL122) Any data previously existing in SMITH's previously defined minidisk space remains unchanged. In addition, data previously existing on the redefined minidisk also remains unchanged.

2. `DIRM FOR DAVE RMD 123 9345 DEVNO 1CB`

The `DIRMAINT RMDISK` command tells `DirMaint` to change DAVE's existing 123 minidisk, whatever its existing device type and location, to be the real 9345 DASD unit on address 1CB. The data on DAVE's existing 123 disk is not changed, but the space may become available for future allocation. The data on device 1CB is not changed. If the volume on 1CB is defined in the `EXTENT CONTROL` file, the otherwise unused space remains allocatable.

## SATELLITE



### Purpose

Use the SATELLITE operand of the DIRMAINT command to control the satellite server machines.

When you specify this command without the FOR and AT prefix keyword options, the command is sent to each of the SATELLITE servers. Use both the FOR and AT keywords to specify the name and node of a SATELLITE server that is running on a different z/VM node from where the DIRMAINT server is running.

### Operands

#### IVP

starts an automated installation verification program run on the specified SATELLITE machine, or obtains status information on the run on the specified SATELLITE machine.

#### START

starts an IVP run. A spool file containing the results of the run is sent to the user that issues the IVP START command. See [“IVP Results”](#) on page 139.

#### STATUS

requests status of the most recent IVP run. A spool file containing the status of the run is sent to the user ID that issues the IVP STATUS command. See [“IVP Results”](#) on page 139.

#### RLDCode

causes the SATELLITE virtual machine to reload its resident operating procedures.

#### RLDData

causes the SATELLITE virtual machine to reload the data tables from the CONFIG\* DATADVH disk files.

**CMS command****CP command**

executes the specified CMS or CP command on the satellite server.

**DIRECT**

requests that the SATELLITE server obtain the current source directory and build a source directory.

**GETCONsole**

specifies a copy of the console logfile from the DIRMAINT machine.

**?**

specifies that the list of available console files is to be displayed to the issuing user.

**FULLdate**

specify for the CP QUERY PRT command.

**ISODate**

specify for the CP QUERY PRT command.

**SHortdate**

specify for the CP QUERY PRT command.

**spoolid**

specifies that a copy of the identified console spool file is to be sent to the command issuer. If *h* is omitted, the currently active console spool file is closed and a copy is sent to the command issuer.

**Note:** Specifying the spool ID of the currently active console spool file will result in an error. To obtain a copy of the currently active console spool file, omit the *spoolod* parameter.

**SHUTDOWN**

requests that the satellite server shutdown.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information about the FULLDATE, ISODATE, SHORTDATE operands, type HELP CP QUERY or see [z/VM: CP Commands and Utilities Reference](#).

## SCAN



Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the SCAN operand of the DIRMAINT command to perform a high speed search of the source directory. A file with all records that match the specified pattern is returned to the requester.

### Operands

#### *pattern*

defines the search pattern.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/ RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The SCAN operand is designed to identify matching directory statements. An asterisk (\*) can be used alone as a wild card indicating that all values found in that position in the record are matches.
3. If only an \* is specified as a pattern parameter, then a match on the corresponding field in each directory statement is unconditional.
4. The search parameter list provided to SCAN is uppercased before the search begins. This must be considered when attempting to scan for directory statements that allow mixed case arguments such as POSIXGROUP and POSIXINFO statements.
5. Any pattern parameter consisting only of wild card designators in the form \*\*\*\*\* is treated the same as a parameter consisting of a single asterisk.
6. A scan pattern consisting of nothing but wild card designators in the form

```
DIRM SCAN * * * * *
is equivalent to the command
```

```
DIRM SCAN *
```

7. The maximum number of scan arguments that are allowed in the pattern is 13.
8. The command `DIRM SCAN *` returns all comment records within the source directory, not all records.
9. Using this operand to scan the source directory may cause sensitive information, logon and minidisk passwords, to be sent to your user ID.

### SCAN Information and Examples

The pattern to be matched should be constructed like the directory entry the user is searching for. The pattern supplied must be the positional equivalent of the directory control statement format the search is targeted for. If you want to search for any users with a maximum logon storage of 16MB you must enter the SCAN operand as follows, because the maximum storage field is the fifth field on the USER statement.

```
DIRM SCAN * * * * 16M
```

The following example will search for all user IDs with a maximum storage value of 16MB. The \* wild card character is not limited to an entire token. It can also be used in conjunction with character strings.

```
DIRM SCAN * * * * 16M
```

```
Userid: <=== Qualifying Record =====>
ALBAHAE USER ALBAHAE password 2M 16M G
ALICEK USER ALICEK password 2M 16M G
AMD USER AMD password 2M 16M G
ANGELA USER ANGELA password 2M 16M G
ANJAVI USER ANJAVI password 2M 16M G
APPINO USER APPINO password 2M 16M G
AP2SVP USER AP2SVP password 2M 16M G
ATORRES USER ATORRES password 2M 16M G
AUSSIM USER AUSSIM password 2M 16M G
BAKIS USER BAKIS password 2M 16M G
BALU USER BALU password 2M 16M G
BARZILA USER BARZILA password 4M 16M G
```

The following example shows a source directory scan based on account number:

```
DIRM SCAN ACCOUNT 4736819
Userid: <=== Qualifying Record ===>
BAP ACCOUNT 4736819 BAP
DENISTO ACCOUNT 4736819 WAYNE
GOODWIN ACCOUNT 4736819 GODWIN
JCOYLE ACCOUNT 4736819
MAINTSY ACCOUNT 4736819 MAINTSY
NAMESRV ACCOUNT 4736819
NSMITH ACCOUNT 4736819 NSMITH
PEDERSN ACCOUNT 4736819 PEDERSN
RLM ACCOUNT 4736819 RLM
ROONEY ACCOUNT 4736819 ROONEY
RUBAS ACCOUNT 4736819 4-032
SDFTST1 ACCOUNT 4736819
SDFTST2 ACCOUNT 4736819
SVM ACCOUNT 4736819 SVM
TCPDBA ACCOUNT 4736819
TESTPRIV ACCOUNT 4736819 TESTPRIV
WIJNEN ACCOUNT 4736819
```

The \* character can be used as a wild card indicating that all values found in that position in the record are matches. For example, to search for all ACCOUNT records with an account number that includes the string DEF, the following command can be issued:

```
DIRM SCAN ACC* *DEF*
```

This command will find all records that contain ACC as the first three characters of the first token in the record and which contain the string DEF anywhere in the second token in the record. Thus it will find any of the following records:

```
ACC      ABCDEF  DISTCODE
ACC      DEFGHI  DISTCODE
ACCOU    BCDEFGHI DISTCODE
ACCOUNT  DEF      DISTCODE
```

It will NOT find the following records:

```
AC      ABCDEF  DISTCODE
ACC     ABC     DEF
ACCOU   ABC     DISTCODE ABCDEF
ACCOUNT ABC     DISTCODE DEF
```

The key to locating the information you are looking for in the source directory is the selection of the scan pattern to be matched. In general, pattern matching occurs in a manner analogous to that performed by the FILELIST command.

Using a wild card designator to obtain a match on specific keyword may not yield the expected results. For example, the minimum abbreviation for the IPL statement is I, but it may also be represented by

*IP* and *IPL*. If you are interested in obtaining a list of what all users in the source directory are *IPL*ing, the selection of a concise scan pattern is not obvious. One possibility is to formulate the scan pattern as follows:

```
DIRM SCAN I*
```

This method will certainly ensure that all *IPL* statements are found, but it has a drawback in that it will also find all *INCLUDE* and *IUCV* statements. Based on the configuration of your source directory, this may cause the output to contain far more information than you wished to see and obscure the information you are interested in. The only way to eliminate extraneous information and be sure that all possible abbreviations are accounted for is to avoid the use of wild card designators and issue three *SCAN* requests, one for each possibility.

To compensate for this flexibility in the way that directory statements can be specified, the *SCAN* processor treats the first pattern parameter in a special manner. The following describes the behavior of *SCAN* processing relative to the first pattern parameter:

1. If the first parameter does not contain an asterisk, that is, if it is not in the form *xxxx\**, *\*xxxx* or *xx\*xx* it is assumed to be the minimum abbreviation of the directory statement in which the invoker is interested. So for the example (*DIRM SCAN I\**) discussed above, the scan pattern that should be supplied to obtain the *IPL* statements from all users' directory definitions is:

```
DIRM SCAN I
```

The output from this invocation will contain all *IPL* statements, regardless of the abbreviation used in the actual statement. The minimum abbreviation for various directory statements can be found in the *DIRECTXA DATADVH* file on the *DIRMAINT* server.

During processing of the source directory, *DirMaint* expands each statement keyword to its full length as it is read. This ensures that regardless of the minimum abbreviation provided, the scan output will contain the records being scanned for. For example, consider the following segment from a directory:

```
USER USER1 SECPASS1
  IN COMMON
USER USER2 SECPASS2
  INC COMMON
USER USER3 SECPASS3
  INCL COMMON
USER USER4 SECPASS4
  INCLUDE COMMON
```

*Scan* will find all of the common records indicated above, regardless of abbreviation, with any of the following command invocations:

```
DIRM SCAN IN
DIRM SCAN INC
DIRM SCAN INCL
DIRM SCAN INCLU
DIRM SCAN INCLUD
DIRM SCAN INCLUDE
```

This does have implications when using a leading asterisk on the first token. The pattern matching is applied against the expanded keyword regardless of the abbreviation found in the source directory.

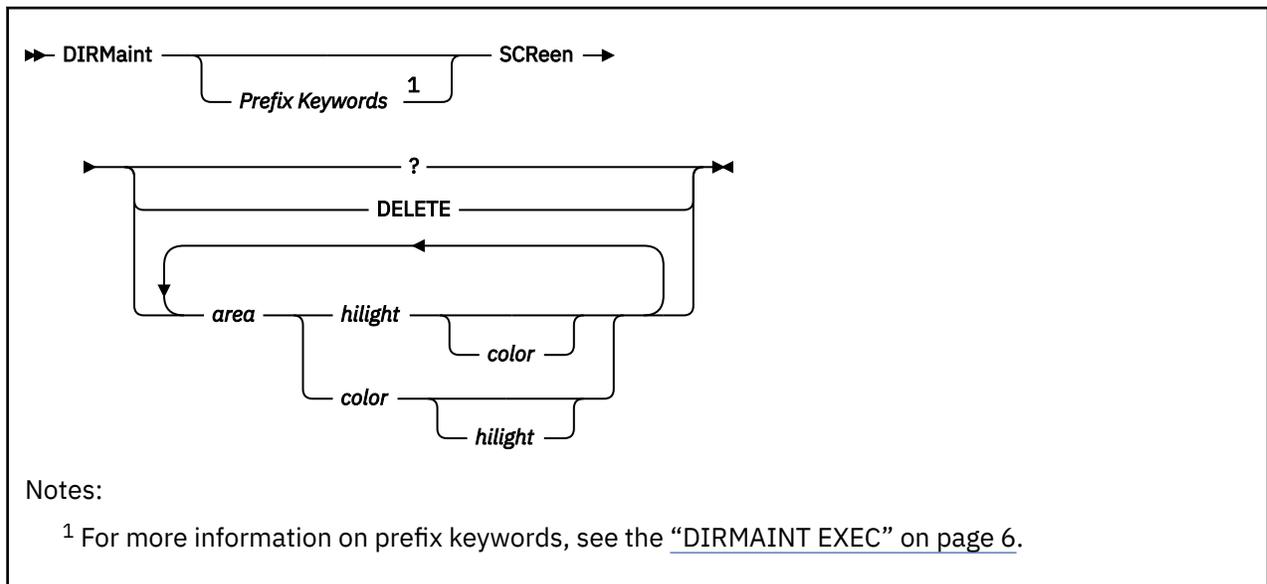
2. If the first parameter is qualified by a suffix asterisk, (that is, of the form *xxx\**), the value comparison is performed between the parameter and the first field on the directory record for the length of the parameter character string up to, but not including, the trailing asterisk. If the first field on the directory record represents a valid abbreviation of a directory statement, the truncated parameter value is compared to the *FULL* statement name for the appropriate length much like the way *FILELIST* evaluates wild card characters.

The implications of this are best explained by the following example. Suppose that the submitted command was:

DIRM SCAN M\*

SCAN processing would return all MACHINE, MDISK, and MINIOPT records in the source directory, since all of these statement types begin with the necessary *character string*.

## SCREEN



### Purpose

Use the SCREEN operand of the DIRMAINT command to assign or change color and extended highlight definition to areas of your virtual machine console display screen. For example, the input, output, and status areas can be assigned specific colors for ease of identification to enhance screen readability. These areas are identified under *area* in the option specifications.

### Operands

**?**

requests a query of the current color and extended highlight definitions for the *area* of the screen.

#### DELETE

requests that the current color and extended highlight definitions for the *area* of the screen be deleted.

#### *area*

specifies the area of the screen. *area* can be:

##### **ALL**

the entire screen.

##### **INArea**

the input area.

##### **STAtarea**

the system status area.

##### **OUTarea**

the output area (CPOut, VMOut, and INRedisp combined).

##### **CPOut**

is the output from the control program.

##### **VMOut**

is the output from CMS or the virtual operating system running in your virtual machine.

##### **INRedisp**

is the input redisplay.

#### *highlight*

specifies the extended highlight value you want to assign to an *area* of your screen. *highlight* can be:

<b>BLInk</b>	blinking
<b>REVvideo</b>	reverse video
<b>UNDerlin</b>	underlining
<b>NONe</b>	no highlighting

**color**

specifies the color attribute you want to assign to an *area* of your screen. The *color* can be:

<b>BLUe</b>	blue
<b>RED</b>	red
<b>GREen</b>	green
<b>YELLow</b>	yellow
<b>TURquois</b>	turquoise
<b>PINK</b>	pink
<b>WHItE</b>	white
<b>DEFault</b>	monochrome (green and white) is the default.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the SCREEN directory statement, see *z/VM: CP Planning and Administration*.
3. Each time you specify the SCREEN operand, you must specify at least one screen *area* option with at least one *color* or *highlight* attribute on the same command line.
4. The operands are processed from left to right and are additive. For example:

```
DIRM SCREEN ALL RED VMO BLUE
```

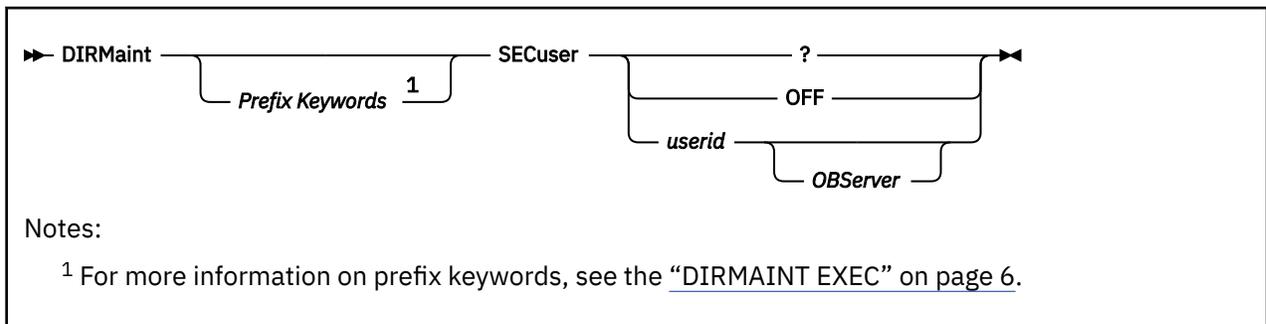
would result in all areas being mapped to RED with the exception of the VMOut area which would be mapped to BLUE.

5. The redundancy rules prohibit specifying an *area* more than once for a specific attribute type. For example, the following specification of the SCREEN operand would be rejected as redundant:

```
DIRM SCREEN ALL RED ALL BLUE
```

6. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## SECUSER



### Purpose

Use the SECUSER operand of the DIRMAINT command to allow the general user to manipulate the secondary console field on their console statement in a directory entry.

### Operands

**?**

queries the directory to see if a secondary user ID is currently specified on their CONSOLE statement.

**OFF**

specifies a request to delete the secondary user ID from their console statement.

**userid**

specifies the user ID to place on the CONSOLE statement to act as the secondary console.

**OBServer**

indicates that the specified user is an observer rather than a secondary user for the virtual machine whose console is being defined.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the CONSOLE directory statement, see *z/VM: CP Planning and Administration*.
3. The use of SECUSER, AUTHSCIF, DSECUSER and DROPSCIF commands are closely related. The *userid* must have granted the issuing user authority through the AUTHSCIF command to use them as a secondary console. An error message will result if this has not been done. For example, if the *userid* WATCHER chooses to become the secondary console of the disconnected user SERVER, this sequence of commands is needed:

- a. WATCHER issues DIRM AUTHSCIF SERVER
- b. SERVER issues DIRM SECUSER WATCHER

When the secondary user facility is no longer needed, one of two commands would be needed:

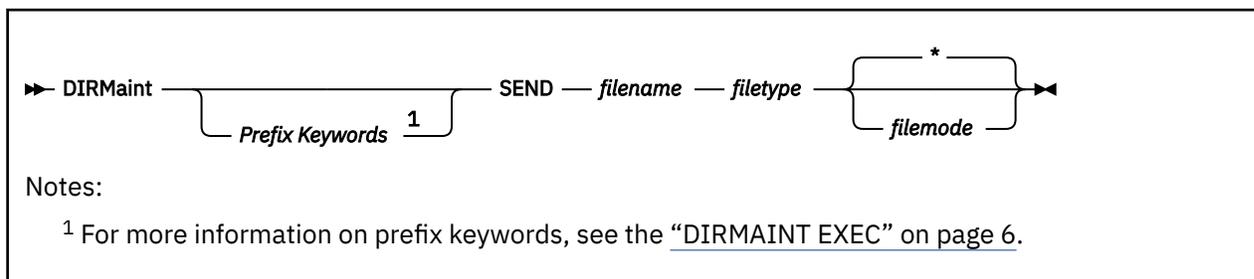
- a. WATCHER issues DIRM DSECUSER SERVER
- b. SERVER issues DIRM SECUSER OFF

If SERVER is not going to use WATCHER as a SCIF user again, the general authority can be cancelled if WATCHER issues:

```
DIRM DROPSCIF SERVER
```

The design allows the SCIF target *userid* (WATCHER) to have overall control of the usage of SCIF, in case of SERVER produced SCIF output which the target was not able to handle. SERVER has sole control of the time it is turned on, but WATCHER must authorize it first. If the SECUSER command is issued before the nominated *userid* has granted authority with the AUTHSCIF command, an error message will result.

## SEND



### Purpose

Use the SEND operand of the DIRMAINT command to request that a copy of a file on one of the DIRMAINT machine's minidisks be sent to your virtual card reader.

### Operands

#### **filename**

specifies the file name of the desired file.

#### **filetype**

specifies the file type of the desired file.

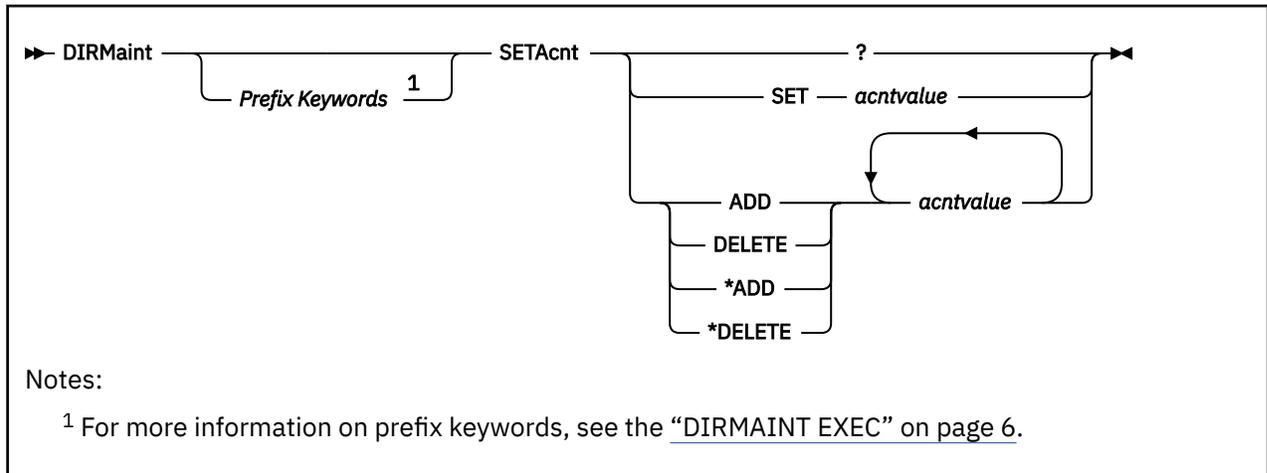
#### **filemode**

specifies the minidisk on which the file resides. If *filemode* is omitted, the default value is \*, or the first file mode in the DIRMAINT machine's search order where the file occurs.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. The requested file will be sent to your virtual card reader. It will be transmitted by the CMS SENDFILE command.

## SETACNT



### Purpose

Use the SETACNT operand of the DIRMAINT command to either SET the primary account number or to manipulate secondary account numbers on an ACCOUNT statement or the \*AC= statement in a profile, user, or identity entry.

### Operands

**?**

requests that the current secondary account settings be returned in a message.

**SET**

requests that the account number that follows be set as primary on the ACCOUNT statement.

**ADD**

**DELETE**

requests that the account numbers that follow be added to, or deleted from, the secondary list of account numbers on the ACCOUNT statement.

**\*ADD**

**\*DELETE**

requests that the account numbers that follow be added to, or deleted from the list of account numbers on the \*AC= statement.

**acctvalue**

the account number (or numbers) to be set, added, or deleted.

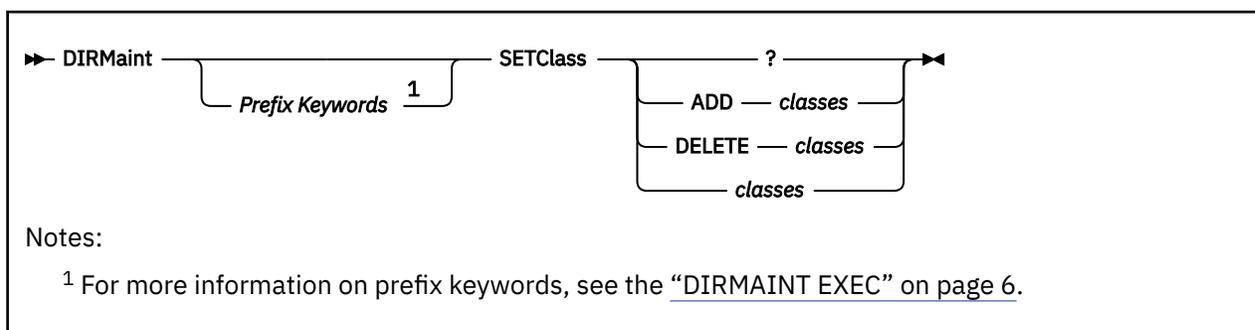
### Usage Notes

1. You may use one or more of the prefix keywords (REQest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the ACCOUNT directory statement, see *z/VM: CP Planning and Administration*.
3. When attempting to add account numbers to the ACCOUNT statement or the \*AC= statement, the request is rejected if the new account numbers already exist on the existing statement.
4. When attempting to add account numbers to the ACCOUNT statement, the request is rejected if the total number of account numbers would exceed the maximum allowed on the ACCOUNT statement.
5. Since the \*AC= statement is a DirMaint construct, it is not bound by the maximums enforced by the CP ACCOUNT directory statement.

## SETACNT

6. When attempting to delete account numbers from the ACCOUNT statement or the \*AC=, the request is rejected if the account numbers being deleted do not exist on the current statement.
7. When attempting to set a primary account number on the ACCOUNT statement, the existing primary account number, if present, will be discarded.

## SETCLASS



### Purpose

Use the SETCLASS operand of the DIRMAINT command to enable you to add, delete, replace, or query the current CLASS value for a profile, user, or identity directory entry.

### Operands

?

allows you to query the current class values.

#### **ADD classes**

#### **DELETE classes**

specifies that the listed class values are to be added to, or deleted from, the existing class values.

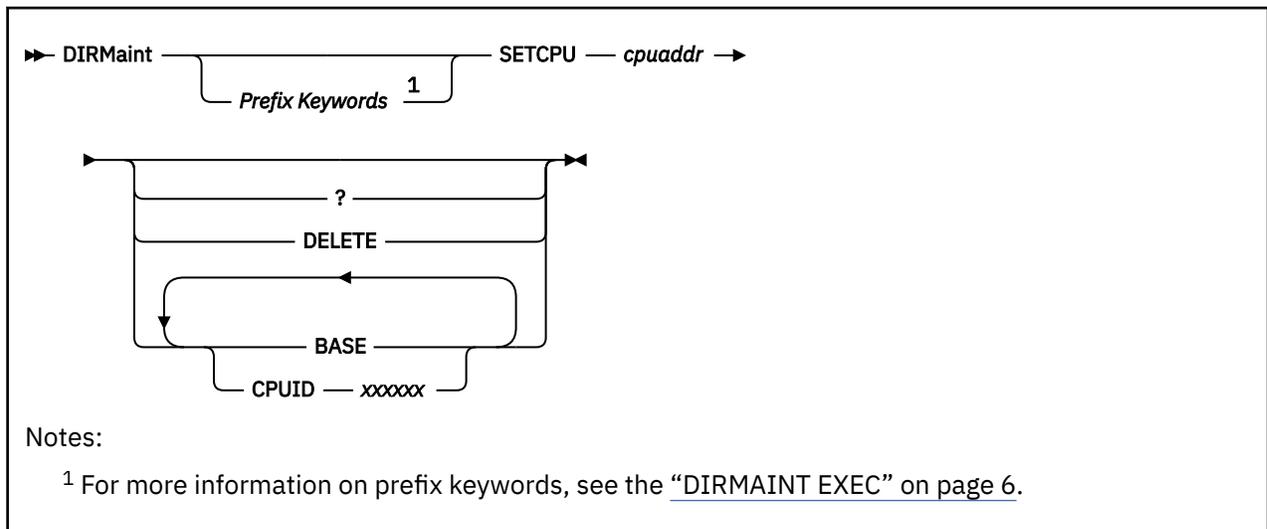
#### **classes**

specified without the ADD or DELETE keyword, the *classes* replace the current class information in the directory entry.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the CLASS directory statement, see [z/VM: CP Planning and Administration](#).
3. If the keyword ADD and DELETE are not specified (that is, only a list of classes is provided) a REPLACE is assumed.
4. An attempt to add a CLASS value that already exists on the USER or IDENTITY statement will be rejected.
5. An attempt to delete a CLASS value from the USER or IDENTITY statement that does not exist will be rejected.

## SETCPU



### Purpose

Use the SETCPU operand of the DIRMAINT command to allow a user to add, change, delete, or query the central processing unit (CPU) statement in a profile, user, or identity entry.

### Operands

#### *cpuaddr*

specifies the address of the virtual processor being defined, changed, or deleted. The processor address can be any hexadecimal value from 0 to 3F.

#### ?

issues a query against the specified CPU statement.

#### DELETE

deletes the specified CPU statement from the directory entry.

#### BASE

specifies that the CPU identified on this statement is defined as the base CPU.

#### CPUID *xxxxxx*

specifies the processor ID number that is stored in bits 8 to 31 of the CPU ID and is returned in response to the STIDP instruction.

### Usage Notes

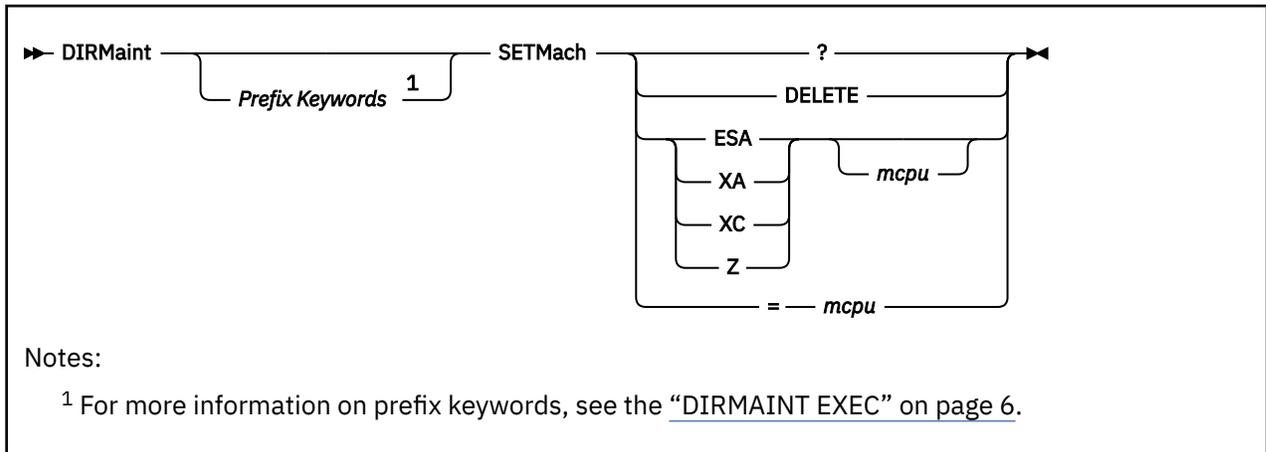
1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the CPU directory statement, see *z/VM: CP Planning and Administration*.
3. Successful execution of the DIRMAINT SETCPU operand depends on the MACHINE directory statement setting defining the maximum number of virtual processors allowed to be defined (indicated by the *mcpu* value). The following rules apply:
  - Modification of an existing CPU statement is always valid. However, only one CPU statement can be given the BASE attribute.
  - Deletion of an existing CPU statement is always valid.
  - The CPU operand is equivalent to the SETCPU operand.

- If a MACHINE statement does exist and the *mcpu* field contains a value, this value is the maximum number of allowable CPU statements.

If any of these rules are violated, SETCPU operand processing will terminate and an appropriate error message is issued.

4. A new CPU statement is always added after all existing CPU statements in your directory entry unless your installation has chosen to sort by device addresses. The order of CPU statements determines the assignment of real processors to virtual CPUs.
5. The CRYPTO operand is still accepted for compatibility purposes, but provides no function.

## SETMACH



### Purpose

Use the SETMACH operand of the DIRMAINT command to allow a user to create, change, or query a MACHINE statement in a profile, user, or identity entry. If a MACHINE statement does not exist in the directory entry, a new statement is created. If a MACHINE statement does exist, the machine type is changed as specified in the SETMACH operand.

### Operands

**?**

requests a query be done of the current MACHINE statement.

**DELETE**

requests that the MACHINE statement be deleted from the directory.

**ESA**

defines an ESA-mode virtual machine.

**XA**

defines an XA-mode virtual machine.

**XC**

defines an XC-mode virtual machine.

**Z**

defines a z/Architecture-only virtual machine.

**=**

allows an existing operand in the specified position to be used in the resulting machine statement.

**mcpu**

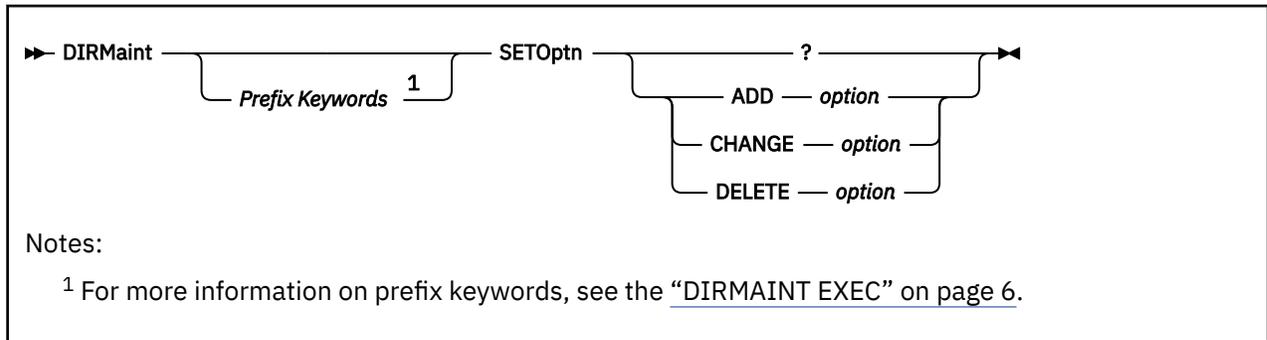
defines the maximum number of CPUs this machine can define.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the MACHINE directory statement, see [z/VM: CP Planning and Administration](#).
3. If the equals (=) operand is used, the *mcpu* operand is required.
4. For more information on the various modes of operation, see the CP SET command in the [z/VM: CP Commands and Utilities Reference](#).

5. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## SETOPTN



### Purpose

Use the SETOPTN operand of the DIRMAINT command to enable you to request that certain CP options be added to, changed on, or deleted from the OPTION control statement in a directory entry. Once the requested options are in the directory control statement, they will be in effect for each subsequent logon. The SETOPTN operand also allows you to query the OPTION control statement in a directory entry.

**Note:** Some options can result in a significant increase in overhead. An installation normally exercises restraint in granting use of these options to the general user.

### Operands

?

requests that the current OPTION statement be returned in a message.

#### ADD

specifies that the option that follows is to be added to the OPTION control statement in the directory entry.

#### CHANGE

specifies that the option that follows is to be changed on the OPTION control statement in the directory entry.

#### DELETE

specifies that the option that follows is to be deleted from the OPTION control statement in the directory entry.

#### *option*

specifies one of these options:

##### **ACct**

specifies that the virtual machine can issue DIAGNOSE code X'4C' to generate accounting records.

##### **AFfinity**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

##### **APplmon**

specifies that the virtual machine can issue DIAGNOSE code X'DC'.

##### **BMX**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

##### **COMsrv**

authorizes the indicated virtual machine to act as a communication server.

##### **CONceal**

places the user in a protected application at logon time.

**CFUSER**

specifies the user is authorized to connect to a Coupling Facility service machine.

**CFVM**

specifies the user is a Coupling Facility service machine. CFVM may not be specified with RMCHINFO.

**CHPIDVirtualization OFF**

CHPID virtualization is not active for the virtual machine. This is the default if CHPIDVirtualization is not included on either the GLOBALOPTS statement or the user's OPTION statement.

**CHPIDVirtualization ONE**

Single path CHPID virtualization is active for the virtual machine.

**CPuid *cpuid***

provides a processor identification (CPUID) in hexadecimal to be stored in response to the STIDP instruction. The *cpuid* value is required for an ADD or CHANGE, but is prohibited for a DELETE.

**CRYMeasure**

authorizes the specified virtual machine to obtain crypto measurement data from the crypto hardware on the real machine. Note that the RMchinfo option should **not** be specified when CRYMeasure is specified.

The CRYMeasure operand is not allowed in a subconfiguration entry.

**DEVInfo**

specifies that the virtual machine can issue DIAGNOSE code X'E4', subcodes X'00' and X'01'.

**DEVMaint**

specifies that the virtual machine can issue DIAGNOSE code X'E4', subcodes X'02' and X'03'.

**DIAG200**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**DIAG88**

specifies that the virtual machine can issue DIAGNOSE code X'88'.

**DIAG98**

specifies that the virtual machine can issue DIAGNOSE code X'98'.

**D84NOPAS**

specifies that the virtual machine can issue all subfunctions of DIAGNOSE code X'84' except LOGPASS and MDISK, without verification of the existing logon password.

**Ecmode**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**IGNMAXU**

indicates that this virtual machine can log on to the system even if the number of users already logged on is equal to or greater than the maximum allowed.

**Isam**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**LANG *langid***

identifies the language that should be set for the virtual machine during logon. The *langid* value is required for an ADD or CHANGE, but is prohibited for a DELETE.

**LINKMSG**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**LKFAC**

indicates that the specified virtual machine is authorized to use the multipath lock facility RPQ simulation support.

**LNKExclu**

specifies that the virtual machine is authorized to use the stable and exclusive access modes.

**LNKNopas**

specifies that the virtual machine is authorized to link to any other virtual machine's DASD without password authorization.

**LNKStabl**

specifies that the virtual machine is authorized to use the stable access modes.

**LXAPP**

specifies the virtual machine can access the SE hard drive using DIAGNOSE X'2C4'. See [z/VM: CP Programming Services](#) for more information on DIAGNOSE X'2C4'.

**MAIntccw**

authorizes the specified virtual machine to use diagnostic CCWs.

**MAXConn *maxno***

specifies the maximum number of IUCV and APPC/VM connections allowed for this virtual machine. The *maxno* value is required for an ADD or CHANGE, but is prohibited for a DELETE.

**MAXVmcfi *maxno***

specifies the maximum number of VMCF inbound messages, including those initiated by SMSG, plus IDENTIFY final response interrupts that may be queued for processing on this virtual machine. The *maxno* value is required for an ADD or CHANGE, but is prohibited for a DELETE.

**MIh**

specifies that CP simulate an interrupt for the virtual machine whenever it detects a missing interrupt condition for an I/O operation.

**NETAccounting**

specifies that Network Data Transmission account records (type 0C) should be generated for this user.

**NETRouter**

specifies that this user is a network router, and Network Data Transmission account records (type 0C) should be generated for this user, and any user that communicates with this user over a virtual LAN.

**NOChain**

Tolerated for migration compatibility from VM/SP HPO. If specified, it will be ignored.

**NOMdcfs**

specifies that the virtual machine can use minidisk cache at a rate that is not limited by the fair share limit.

**NOMEMAssist**

Specifies that MEMASSIST support is disabled at logon time. Without this option, MEMASSIST support is enabled by default at logon time if the facility is available on the system.

**NOVF**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**Quickdsp**

causes a virtual machine to be added to the dispatch list immediately when it has work to do.

**REaltimer**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**RMchinfo**

indicates that the specified virtual machine is authorized to access real-machine configuration information, without regard to the virtual machine's configuration. It should be used when OSA/SF (or any other application that may require real machine configuration information) is running under CMS. RMCHINFO should not be specified for guest operating systems.

The RMCHINFO operand is not allowed in subconfiguration entries.

**SEorig**

specifies that the virtual machine can issue DIAGNOSE X'F8', subfunction X'00'.

**STFirst**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**STGexempt**

specifies that the virtual machine is exempt from CP free storage limit detection.

**STHYI-Util**

authorizes the virtual machine to invoke utilization related functions of the Store Hypervisor Information (STHYI) instruction.

This operand is not allowed in subconfiguration entries.

**STHYI-Guest**

authorizes the virtual machine to invoke guest related functions of the Store Hypervisor Information (STHYI) instruction.

This operand is not allowed in subconfiguration entries.

**STHYI-Respool**

authorizes the virtual machine to invoke resource pool related functions of the Store Hypervisor Information (STHYI) instruction.

This operand is not allowed in subconfiguration entries.

**SVCaccl**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**SVCoff**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**SVC76VM**

specifies that errors are not recorded by CP.

**SVMstat**

specifies that the virtual machine is a service virtual machine. The monitor data records associated with this virtual machine include the SVMSTAT setting. The only purpose is to allow products that process monitor data to report on service virtual machines separate from end-user virtual machines. No other operations, such as transaction or wait state classification, are affected by this operand.

The SVMstat operand is not allowed in subconfiguration entries.

**Todenable**

specifies that the user may change the virtual machine's time-of-day clock with the SCK instruction.

**VCunoshr**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**VMsave**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

**XMEM**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

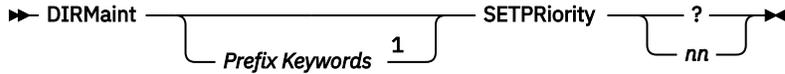
**370e**

This operand is maintained for compatibility purposes only. It will be accepted without error, but otherwise ignored.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. The DirMaint Release 4 SETOPTN command has been renamed to the USEROPTN command for DirMaint Release 5. For more information, see the USEROPTN command. The SETOPTN command in level 140A performs the same function as in DirMaint Release 4. For level 150A, the SETOPTN command allows the privileged user to add or change the values in the OPTION directory statement.
3. Not every option is valid for every release or every directory entry type. For more information on the OPTION directory statement, see [\*z/VM: CP Planning and Administration\*](#).

## SETPRIORITY



### Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the SETPRIORITY operand of the DIRMAINT command to query or change the CP dispatching priority in a user or identity directory entry.

### Operands

**?**

requests that the current priority be displayed.

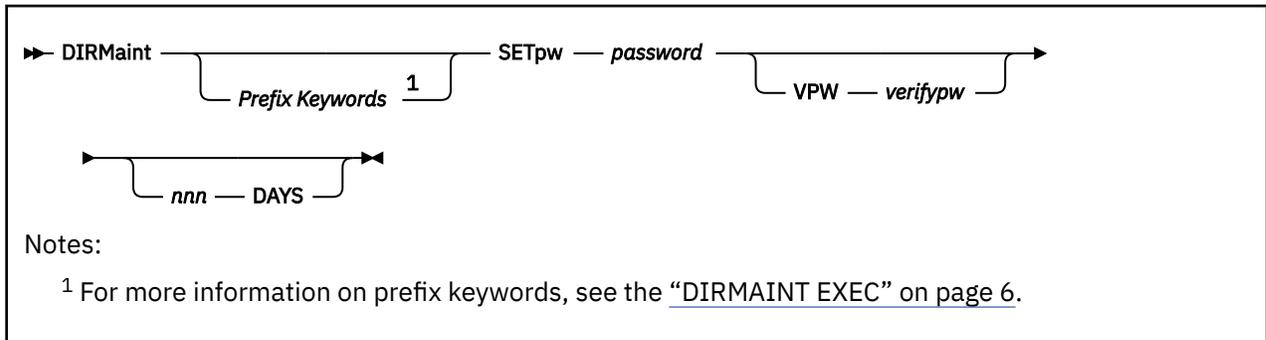
**nn**

is a two-digit decimal number that is to be the new CP dispatching priority. The highest priority that can be given a user is 01, the lowest (slowest) is 99. The default is 64.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. PRIOSet and PRIORITY are synonymous with SETPRIORITY. SETPRIORITY is the preferred operand. The others are retained for compatibility with previous releases.
3. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## SETPW



### Purpose

Use the SETPW operand of the DIRMAINT command to change the CP logon password for a user.

### Operands

#### *password*

specifies the value to be used as a password or passphrase for the user. When entering the SETPW command password or passphrase on the VM command line, note that a passphrase containing embedded blanks must be surrounded by single quotes (with single quote characters within the passphrase itself doubled). When entering the password or passphrase on the SETPW command menu panel, up to 65 characters may be used. Longer passphrases must be specified by entering the DIRMaint command on the VM command line or by issuing the DIRMaint command in a z/VM Rexx EXEC. The password or passphrases will not appear in DirMaint maintained log files.



**Attention:** Use of this parameter may show up in console trace logs if tracing is turned on.

#### **VPW** *verifypw*

specifies the same password again for typographical verification.

#### *nnn* **DAYS**

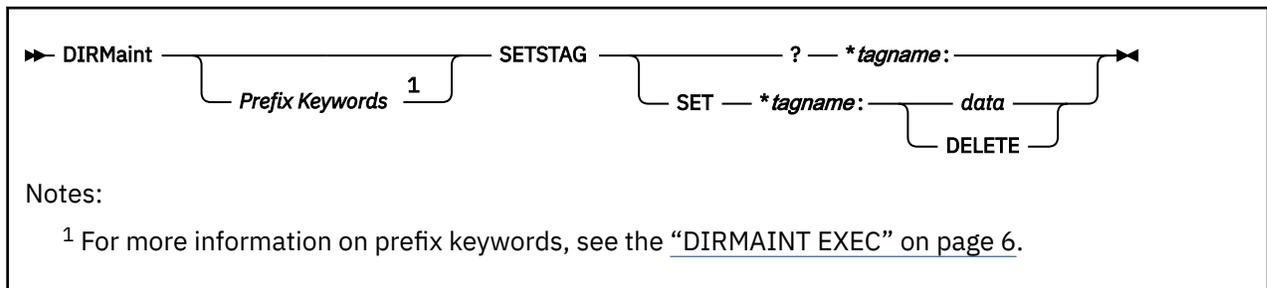
specifies the number of days till the password expires. *nnn* can be any decimal number from 0 to 366. If not specified, the CONFIG\* DATADVH file is consulted and the value specified on the PW\_INTERVAL\_FOR\_SET= entry is used.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Your installation may have rules about password content, minimum length, required frequency of change, or reuse of old passwords. These can be enforced by the PASSWORD\_SYNTAX\_CHECKING\_EXIT exit routine. For more information, see *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*. for information on the PASSWORD\_SYNTAX\_CHECKING\_EXIT.
3. A logon password of NOLOG can be set using the SETPW command to prevent a user ID from logging on.
4. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84') if it is enabled.

5. If your system specifically has RACF installed as the ESM, see the [\*z/VM: Directory Maintenance Facility Tailoring and Administration Guide\*](#) for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## SETSTAG



### Purpose

Use the SETSTAG operand of the DIRMAINT command to assign a value to a locally defined tag within a directory entry. If the value is already set it may also be queried.

### Operands

**?**

requests that the current setting of the specified tag be returned in a message.

**\*tagname:**

is a locally defined tagname.

**SET**

requests that the specified tag be changed or added.

**data**

specifies the data to be assigned to the local tag.

**DELETE**

indicates that the tag should be deleted from the directory.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. If *data* is provided, its case is respected when placing into the directory.
3. Due to the way DirMaint processes local tags, extra spaces between words are reduced to a single space in local tag data.
4. Data that exceeds a single directory record is split and represented on two lines in the directory entry.
5. The widespread use of local tags has the ability of significantly expanding the size of the source directory. This should be kept in mind when defining local tags and authorizing users to use them.
6. A length limitation is enforced on the data being assigned to the user defined tag. A limit of 60 characters must not be exceeded when the tag name (which is of variable length) is combined with the longest blank delimited token by a single blank. For example, if the stag was:

```
*MYSTAG:
```

And the incoming data is:

```
This is my
Data 123456789!123456789!123456789!123456789!123456789!123456789!
```

The combination of the tag name and the longest token

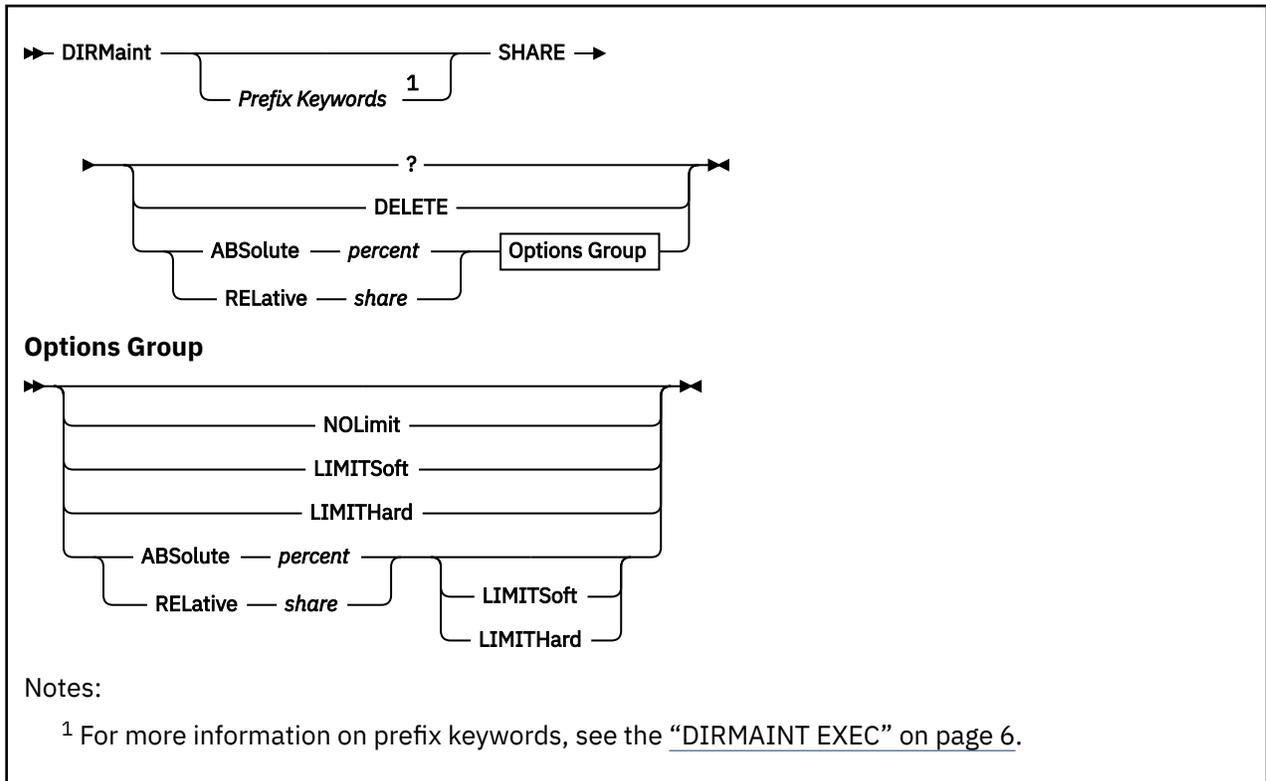
```
*MYSTAG: 123456789!123456789!123456789!123456789!123456789!123456789!
```

Would exceed 60 characters and the request would be rejected. If the new data was:

```
This is my  
Data 123456789 123456789 123456789 123456789 123456789 123456789
```

The request would be accepted since no single token would cause the length to exceed 60 characters when combined with the tag name.

## SHARE



### Purpose

Use the SHARE operand of the DIRMAINT command to manipulate the CP SHARE directory statement. This statement specifies a virtual machine's share of CPU power.

### Operands

**?**

requests that the current share settings be returned in a message.

#### DELETE

requests that the current share statement be deleted.

#### ABSolute

specifies a minimum or maximum absolute share of all active processors in a system.

#### *percent*

a real number with at most one decimal place, within the range .1 to 100 followed by a percent sign. For example, 22.9%.

**Note:** The decimal place following the value may not be valid on some releases. Consult your system documentation to obtain specific information for your release.

#### RELative

specifies a minimum or maximum relative share of all active processors in a system.

#### *share*

an integer number within the range from 1 to 10000.

#### NOLimit

specifies that a user's share of processing resource is not limited.

**LIMITSoft**

specifies that the share of processing resource is limited but at times these users will receive more than their limit.

**LIMITHard**

specifies that the share of processing resource is limited. These users will not receive more than their limit.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the SHARE directory statement, see [\*z/VM: CP Planning and Administration\*](#).
3. When both the minimum value and maximum value for ABSolute or RELative are specified in the same invocation, the minimum value must not exceed the maximum value.

## SHUTDOWN

```
►► DIRMaint ———— SHUTDOWN ◄◄  
                |  
                | Prefix Keywords 1  
                |
```

Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

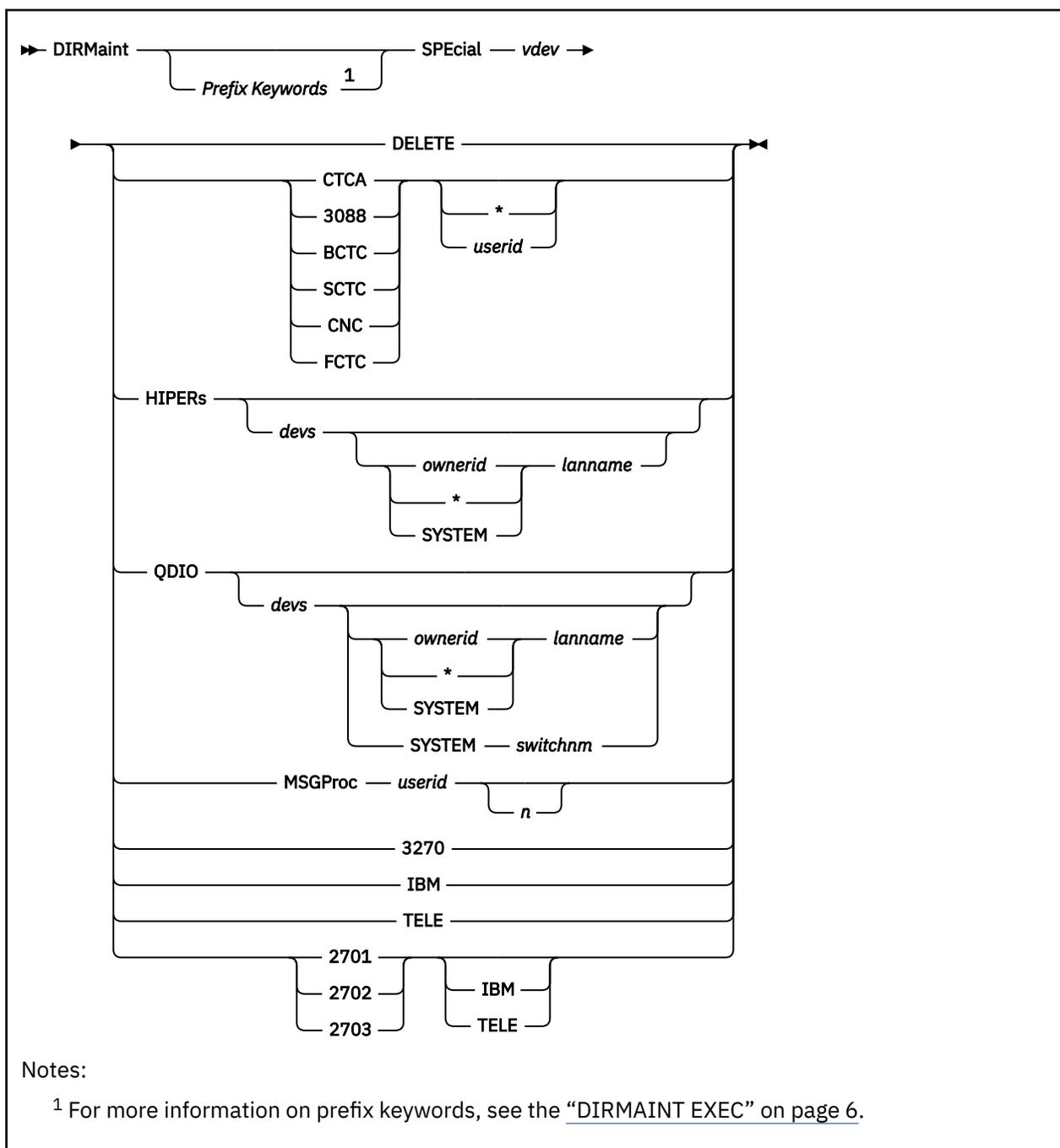
### Purpose

Use the SHUTDOWN operand of the DIRMAINT command to shutdown the DIRMAINT virtual machine in an orderly manner.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. This operand should be invoked a few minutes before a complete shutdown of the system. This gives the DIRMAINT virtual machine time to shutdown in an orderly manner and to log itself off.
3. Any operand currently running will complete normally, and then the DIRMAINT virtual machine will log off if the DIRMAINT machine is running disconnected when the command is received. If the DIRMAINT machine is running connected, the machine will restart and leave the console at the ready prompt. Any requests still queued on disk will remain queued and will be executed after the next startup.

## SPECIAL

**Purpose**

Use the SPECIAL operand of the DIRMAINT command to allow the user to add or delete a SPECIAL directory statement to a directory entry.

**Operands**

## SPECIAL

### **vdev**

For most SPECIAL devices, this is the virtual device address of the device to be defined. For a MSGPROC, HIPERS, or QDIO device, this represents the base (first) device address in a series of virtual I/O devices that belong to the same unit.

A simulated adapter can also be created using the NICDEF statement, which allows additional configuration options not available on the SPECIAL statement. See “NICDEF” on page 173 for more information.

### **DELETE**

requests that the SPECIAL statement be deleted.

### **CTCA**

### **3088**

specifies either CTCA or 3088 for a virtual 3088 Multisystem channel Communication Unit logical channel adapter.

### **BCTC**

### **SCTC**

### **CNC**

### **FCTC**

specifies either a BCTC, SCTC, CNC, or FCTC virtual communications adapter.

### **\***

tells CP that coupling is to be allowed only from another virtual communications adapter owned by the same virtual machine that owns the virtual communications adapter defined by this SPECIAL statement.

### **userid**

identifies the user ID of a virtual machine allowed to connect to this virtual communications adapter using the CP COUPLE command.

### **HIPERS**

indicates that a simulated HiperSockets adapter should be created based on this statement. A simulated Network Interface Card (NIC) is defined during LOGON with *devs* devices (beginning with the base, *vdev*). If a VM LAN is identified, the NIC is automatically coupled to *ownerid lanname*.

### **devs**

specifies the number (decimal) of virtual I/O devices to be created for this adapter. For a simulated HiperSockets adapter, *devs* must be a decimal value between 3 and 3,072 (inclusive). If the *devs* value is omitted, the default for HiperSockets is three (3) devices.

### **ownerid|\*|SYSTEM lanname**

identifies a virtual LAN segment for an immediate connection. When *ownerid* is specified as asterisk (\*) it is resolved as the *userid* of the current virtual machine. When *ownerid* and *lanname* are omitted, the HiperSocket adapter is left in the default (uncoupled) state. When *ownerid* and *lanname* are specified, the adapter is connected to the designated virtual LAN segment automatically.

*ownerid* may be specified as SYSTEM, indicating that the virtual LAN segment is system-owned.

### **QDIO**

indicates that a simulated QDIO adapter should be created based on this statement. A simulated Network Interface Card (NIC) is defined during LOGON with *devs* devices (beginning with the base device, *vdev*). If a VM LAN or z/VM Virtual Switch is identified, the NIC is automatically coupled to the specified *lanname* or *switchnm*.

### **devs**

specifies the number (decimal) of virtual I/O devices to be created for this adapter. If the *devs* parameter is omitted, the default for a QDIO adapter will be (3). The maximum value is 240.

### **ownerid|\*|SYSTEM lanname**

identifies a virtual LAN segment for an immediate connection. When *ownerid* is specified as asterisk (\*) it is resolved as the *userid* of the current virtual machine. When *ownerid* and *lanname*

are omitted, the QDIO adapter is left in the default (uncoupled) state. When *ownerid* and *lanname* are specified, the adapter is connected to the designated virtual LAN segment automatically.

*ownerid* may be specified as SYSTEM, indicating that the virtual LAN segment is system-owned.

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

identifies a z/VM Virtual Switch for an immediate connection. When SYSTEM *switchnm* is omitted, the QDIO adapter is left in the default (uncoupled) state. When SYSTEM *switchnm* is specified, the adapter is connected to the designated z/VM Virtual Switch automatically.

#### **MSGProc *userid n***

defines a virtual message processor and *n* associated message devices in the virtual I/O configuration. It creates a message facility environment for the user and establishes a connection to the specified Coupling Facility (CF) Service Machine supplying the message path function. The virtual message processor will only be defined if these conditions exist:

- OPTION CFUSER or OPTION CFVM must be specified in the directory entry for this usage. The *vdev* specified must be the first of four available consecutive device numbers in the user's virtual configuration.
- The CF Service Machine user ID specified must be running prior to this virtual machine.
- The *n* specifies the number of message devices that will be created. The device numbers will be *vdev*, *vdev+1*, and so forth up to *vdev+n-1*.

If this SPECIAL MSGPROC statement appears in a:

- CFUSER directory entry, this must be in the range of 4 to 16, and must be even. This defaults to 4. If the CFUSER is a 390-mode virtual machine, the *n* you specify will be ignored and 4 will be used instead.
- CFVM directory entry, this must be in the range of 2 to 8. This defaults to 2. If the CFVM is a 390-mode virtual machine, the *n* you specify will be ignored and 2 will be used instead.

#### **3270**

is the value for the 3270 display device.

#### **IBM**

#### **TELE**

is the virtual device type of the line you are defining.

#### **2701**

#### **2702**

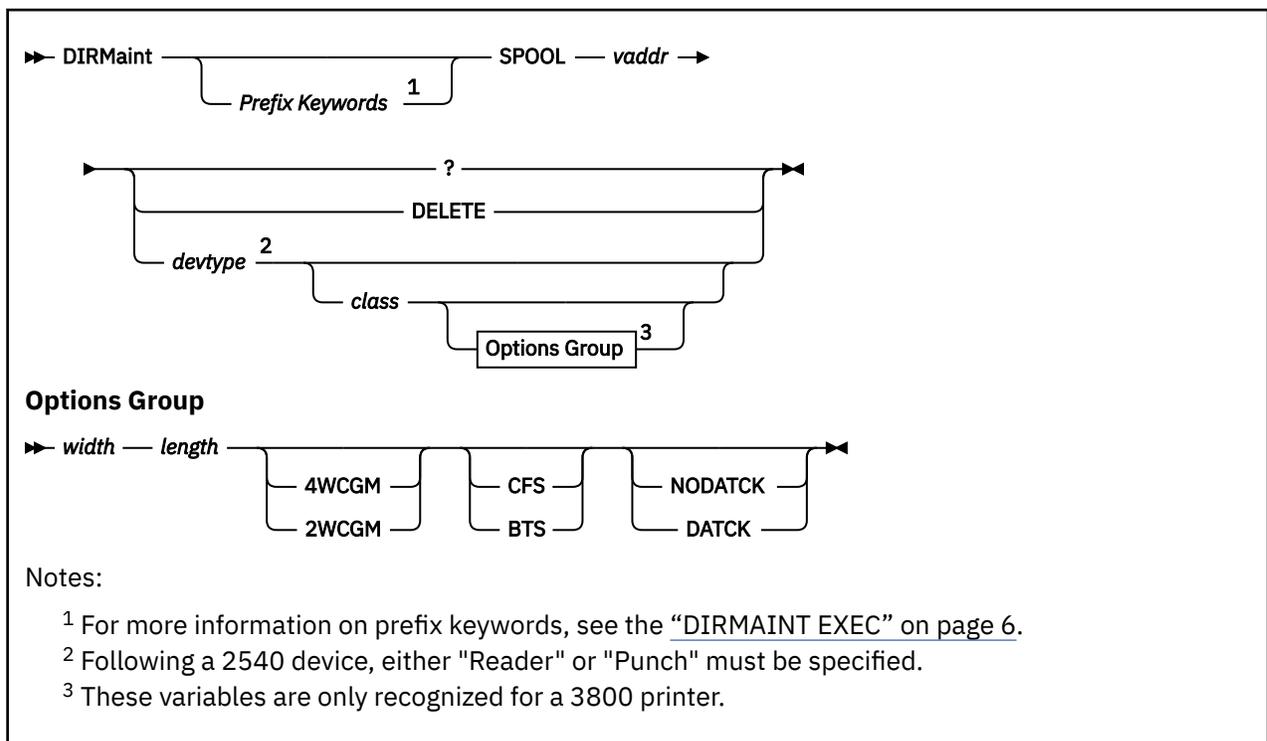
#### **2703**

is the value for a communication line.

### **Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. All options may not be valid for all releases of VM. For more information on the SPECIAL directory statement, see [z/VM: CP Planning and Administration](#).

## SPOOL



### Purpose

Use the SPOOL operand of the DIRMAINT command to add or delete virtual unit record device statements.

### Operands

#### **vaddr**

specifies the virtual device number for the spooling device.

#### **devtype**

specifies the device type. The valid device types are:

- PCH
- PUnch
- Printer
- PRT
- RDR
- Reader
- VAFP
- 1403
- 2501
- 2540
- 3203
- 3211
- 3262
- 3289

- 3505
- 3525
- 3800
- 3800-1
- 3800-3
- 4245
- 4248

**class**

a 1 digit alphanumeric character from A to Z, from 0 to 9, or an asterisk (\*) that indicates the spooling class. An asterisk (\*) is only valid for a reader device.

**width length**

specifies the physical characteristics of the paper to be loaded into the 3800 printer. The variable *width* is the hexadecimal form width code of the paper and *length* indicates the decimal length of the paper.

**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 scan elements of one character set. A 3800-1 can have either two or four WCGMs. A 3800-3 has four WCGMs.

**BTS****CFS**

specifies the type of stacker for the virtual 3800 printer. You may specify either CFS (continuous forms stacker) or BTS (burster trimmer stacker).

**DATCK****NODATCK**

specifies whether CP processes certain virtual 3800 data checks for the virtual machine.

**Note:** Specifying DATCK severely increases overhead associated with simulation of Write and Skip CCWs to the virtual 3800. Specify DATCK only when absolutely necessary.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the SPOOL directory statement, see [z/VM: CP Planning and Administration](#).

## SSI

►► DIRMaint ——— SSI — *nodeID* ◄◄  
                   └── Prefix Keywords 1 ─┘

Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the SSI operand of the DIRMAINT command to prepare a source directory to be used on a node in a single system image (SSI) cluster. This operand updates the DIRECTORY statement with the SSI keyword, removing all other currently existing options, and changing all BUILD ON \* statements in the source directory to BUILD ON *nodeID*.

### Operands

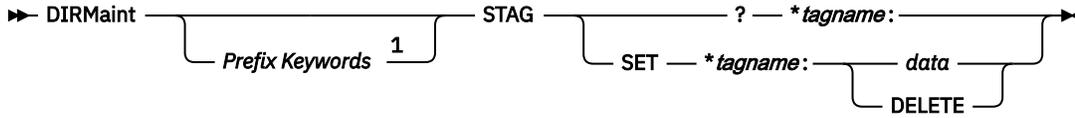
#### *nodeID*

specifies the node in an SSI cluster, to be used in BUILD statements.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords do not apply to this command. They are accepted without error, but otherwise ignored.
2. This operand is allowed in an SSI cluster only.

## STAG



Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC” on page 6](#).

### Purpose

Use the STAG operand of the DIRMAINT command to assign a value to a locally defined tag within a directory entry. If the value is already set it may also be queried.

### Operands

**?**

requests that the current setting of the specified tag be returned in a message.

**\*tagname:**

is a locally defined tagname.

**SET**

requests that the specified tag be changed or added.

**data**

specifies the data to be assigned to the local tag.

**DELETE**

indicates that the tag should be deleted from the directory.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. If *data* is provided, its case is respected when placing into the directory.
3. Due to the way DirMaint processes local tags, extra spaces between words are reduced to a single space in local tag data.
4. Data that exceeds a single directory record is split and represented on two lines in the directory entry.
5. The widespread use of local tags has the ability of significantly expanding the size of the source directory. This should be kept in mind when defining local tags and authorizing users to use them.
6. A length limitation is enforced on the data being assigned to the user defined tag. A limit of 60 characters must not be exceeded when the tag name (which is of variable length) is combined with the longest blank delimited token by a single blank. For example, if the stag was:

```
*MYSTAG:
```

And the incoming data is:

```
This is my
Data 123456789!123456789!123456789!123456789!123456789!123456789!
```

The combination of the tag name and the longest token

```
*MYSTAG: 123456789!123456789!123456789!123456789!123456789!123456789!
```

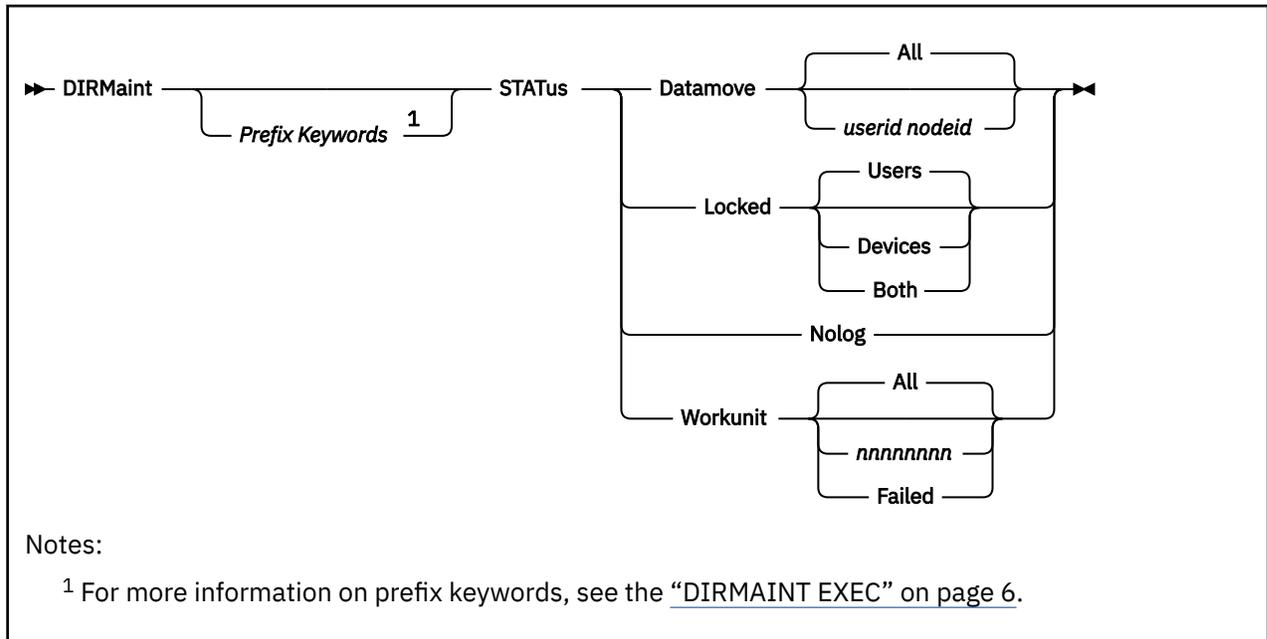
## STAG

Would exceed 60 characters and the request would be rejected. If the new data was:

```
This is my  
Data 123456789 123456789 123456789 123456789 123456789 123456789
```

The request would be accepted since no single token would cause the length to exceed 60 characters when combined with the tag name.

# STATUS



## Purpose

Use the STATUS operand of the DIRMAINT command to request various status information about the DIRMAINT or DATAMOVE virtual machines.

## Operands

### Datamove

specifies a request for the current status of the DATAMOVE virtual machine(s).

### Locked

specifies a request for a list of all directory entries which are locked (to prevent simultaneous updates).

**Note:** Directory entries are locked by a specific invocation of the LOCK or GET operands and can only be unlocked by the UNLOCK or REPLACE operands.

### Users

requests that all locked directory entries be reported. This is the default.

### Devices

requests that all locked devices be reported.

### Both

requests that all locked directory entries and devices be reported.

### Nolog

specifies a request for a list of those user IDs whose logon passwords are NOLOG.

### userid nodeid

requests that a specific DATAMOVE machine be listed.

### Workunit

requests a status a Work Unit Control Files (WUCF) from the DASD control subsystem.

### All

requests that all current active WORKUNITS or DATAMOVE machines be listed. This is the default.

**nnnnnnnn**

requests the status of a specific WORKUNIT (identified by the passed eight digit identifier) be reported. This value may represent an ACTIVE or FAILED workunit.

**Failed**

requests that a list of all failed WORKUNIT's be reported.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. A directory entry is locked to prevent concurrent updates to the source while an administrative activity is underway. A directory entry can be locked with an explicit DIRM LOCK command or by a DIRM GET (without the NOLOCK option). The directory entry is unlocked when a DIRM REPLACE command is processed or by an explicit DIRM UNLOCK command.
3. A device is locked to indicate a pending activity on the device. An example of this is a pending AMDISK request for a minidisk address. The device will be unlocked when the AMDISK request completes successfully (or when it is rolled back during a failure). The lock may remain active if an AMDISK fails and DirMaint was unable to roll back the transaction. The device may be unlocked by an explicit DIRM UNLOCK *addr* command.
4. The existence of a failed WORKUNIT indicates that a work unit was created for a task that could not be accomplished. DIRMAINT will take steps to *roll back* any changes made by this work unit under many circumstances. The Work Unit Control File is maintained for detailed study by the system administrator. The contents of the failed workunit can be determined using the WORKUNIT option of this command. The failed workunit is maintained on the primary directory disk (the 01DF by default) on the DIRMAINT server as:

```
nnnnnnnn WUCFFAIL
```

Active workunits are maintained on the same disk on the DIRMAINT server as:

```
nnnnnnnn WORKUNIT
```

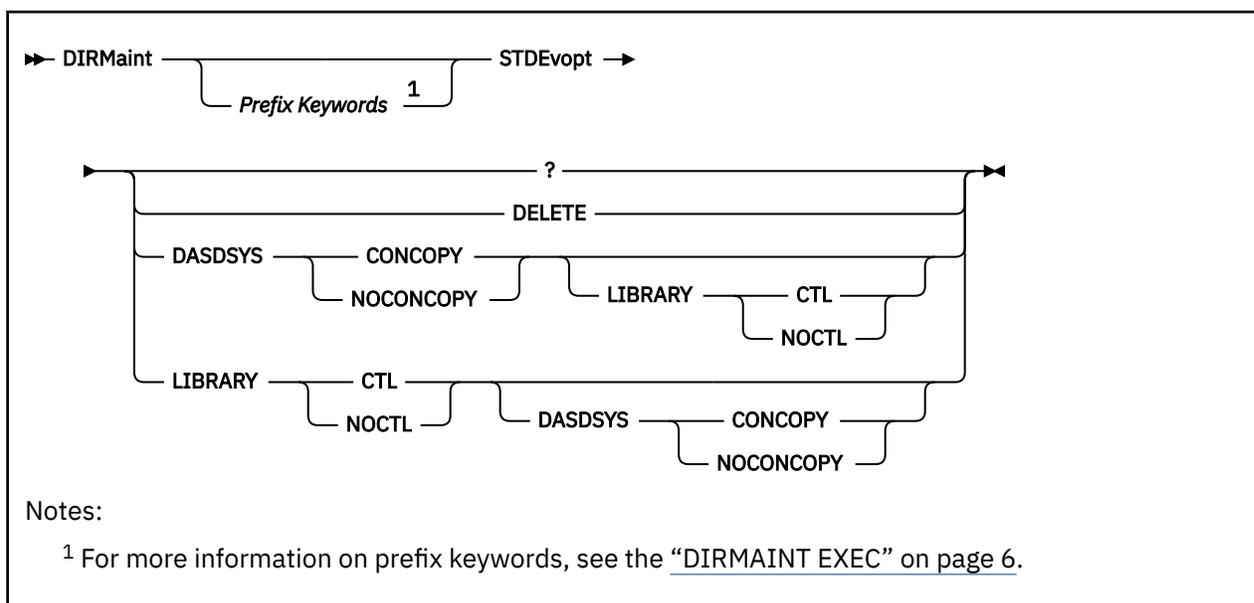
Where:

**nnnnnnnn**

specifies the workunit number in both cases.

5. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.

## STDEVOPT



### Purpose

Use the STDEVOPT operand of the DIRMAINT command to allow you to specify the optional storage device management functions available to a virtual machine. The STDEVOPT operand is applicable to profile, user and identity directory entries.

### Operands

?

requests that the current STDEVOPT directory control statement specifications for the virtual machine be returned in a message.

#### DELETE

deletes any STDEVOPT directory entries for the virtual machine.

#### DASDSYS CONCOPY

#### DASDSYS NOCONCOPY

tells CP whether the virtual machine is authorized to control and process Concurrent Copy and Peer To Peer Remote Copy Establish Pair CCW.

#### LIBRARY CTL

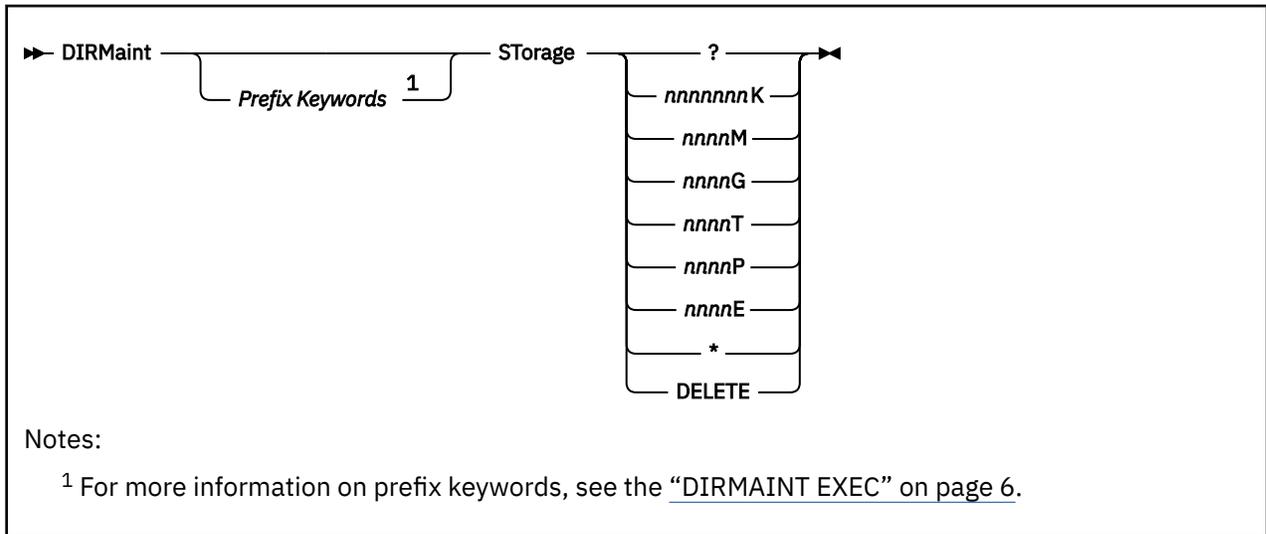
#### LIBRARY NOCTL

tells CP whether the virtual machine is authorized to control a 3495 Tape Library Dataserver.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the STDEVOPT directory statement, see [z/VM: CP Planning and Administration](#).

## STORAGE



### Purpose

Use the STORAGE operand of the DIRMAINT command to display the default and maximum virtual machine storage sizes or to change the default storage size on the USER or IDENTITY statement.

### Operands

**?**

specifies that this transaction is a query to determine the current maximum virtual storage size for this user.

**nnnnnnnK**

specifies the default storage value in kilobytes.

**nnnnM**

specifies the default storage value in megabytes.

**nnnnG**

specifies the default storage value in gigabytes.

**nnnnT**

specifies the default storage value in terabytes.

**nnnnP**

specifies the default storage value in petabytes.

**nnnnE**

specifies the default storage value in exabytes.

**\***

specifies when used within a:

- User or identity directory entry, the value specified within the included profile entry will be used if present, or the CP default value of 1M will be used if there is no STORAGE value within the included profile; and
- Profile directory entry, the CP default value of 1M will be used unless the including user or identity entry specifies an overriding STORAGE value.

**DELETE**

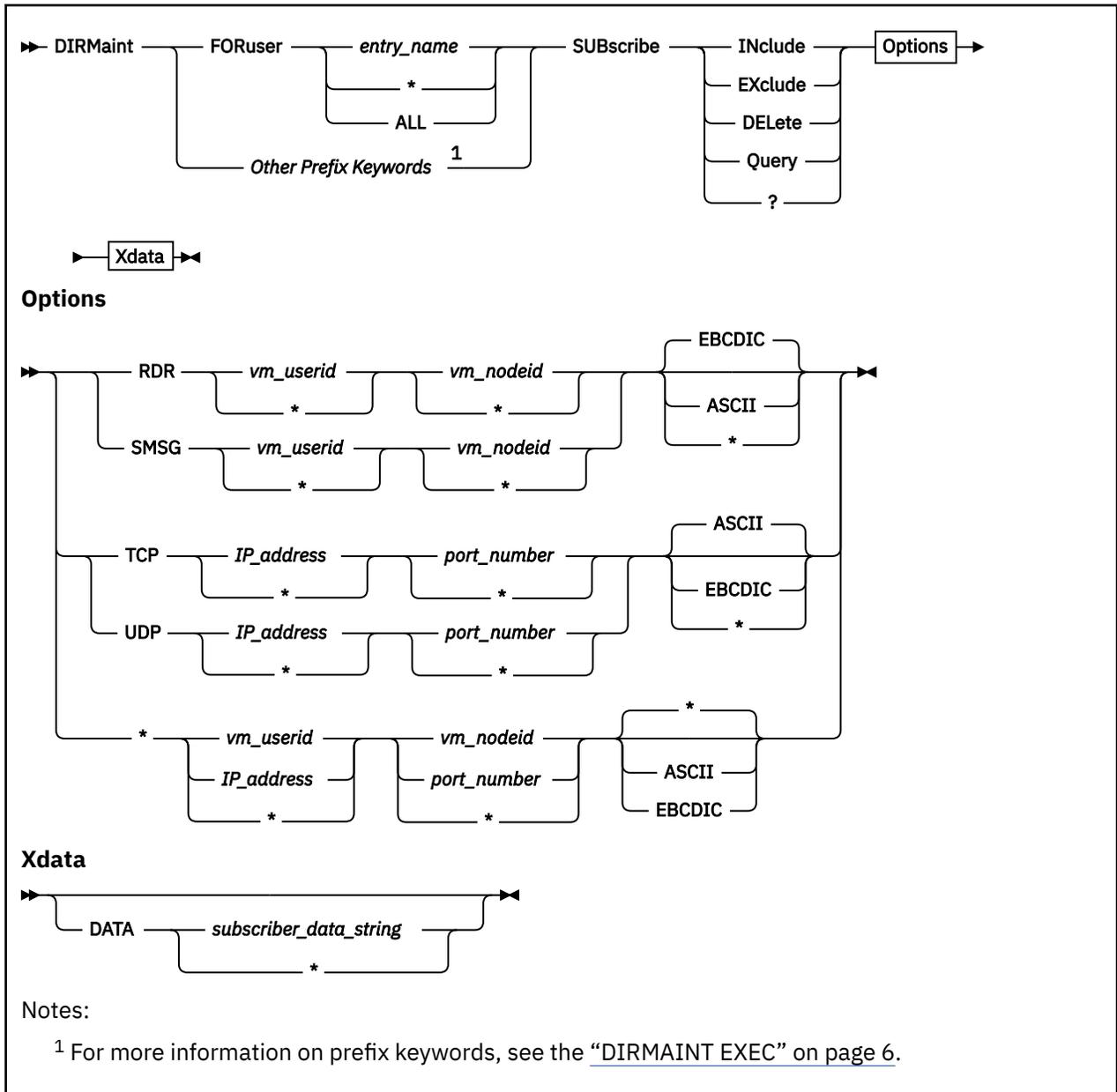
specifies that the STORAGE statement will be deleted, resulting in use of the:

- Value specified within an included profile if deleted from a user or identity directory entry, or use of the CP default value of 1M if there is no STORAGE value within the included profile
- CP default value of 1M when deleted from a profile entry, unless the including user or identity entry specifies an overriding STORAGE value.

## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the STORAGE directory statement, see [z/VM: CP Planning and Administration](#).
3. The suffix K, M, G, T, P or E is required.
4. A query request will report the default values for the default storage size and maximum storage size if they are not specified on the USER or IDENTITY statement.
5. The value is placed into the directory as provided once it has been validated for correct format. Any required rounding is done by CP.
6. If the directory definition of the user ID being modified does not currently have a maximum virtual storage sizes defined, a default maximum storage size of 1MB is assumed. This value becomes the maximum that can be specified for the default virtual storage size.
7. The values supplied must be whole numbers. Values such as 4096.8K or 6.5M are not supported.
8. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.
9. The value specified with this command will be added to the corresponding field in the USER or IDENTITY statement.
10. Use the DirMaint operand MAXSTORAGE to change the maximum storage size on the USER or IDENTITY statement.

# SUBSCRIBE



## Purpose

Use the SUBSCRIBE operand of the DIRMAINT command to manage subscriptions to directory updates. Specifically, you can add new subscriptions, delete old subscriptions, and query existing subscriptions.

## Operands

### *entry\_name*

is the name of the profile, user, identity, or subconfiguration entry subscribed to, or ALL. In the case of DELETE and QUERY requests, an asterisk will be considered ALL. In the case of INCLUDE and EXCLUDE requests, an asterisk will be considered the userid for whom the command is issued.

**RDR**  
**MSG**  
**TCP**  
**UDP**

specifies the protocol. In the case of DELETE and QUERY requests, an asterisk will match all entries. In the case of INCLUDE and EXCLUDE requests, an asterisk is not allowed.

***vm\_userid***  
***IP\_address***

is the first part of the destination parameter (*vm\_userid* for RDR and MSG, *IP\_address* for TCP and UDP). In the case of DELETE and QUERY requests, an asterisk will match all entries.

***vm\_nodeid***  
***port\_number***

is the second part of the destination parameter (*vm\_nodeid* for RDR and MSG, *port\_number* for TCP and UDP). In the case of DELETE and QUERY requests, an asterisk will match all entries. An asterisk is allowed for INCLUDE or EXCLUDE requests for RDR or MSG, but such a subscription cannot be separately deleted or queried.

**ASCII**  
**EBCDIC**

specifies the encoding. EBCDIC is the default for RDR and MSG, and ASCII is the default for TCP and UDP. In the case of DELETE and QUERY requests, an asterisk will match all entries. In the case of INCLUDE and EXCLUDE requests, an asterisk is not allowed.

***subscriber\_data\_string***

is an optional character or hexadecimal string supplied by the subscriber. In the case of DELETE and QUERY requests, an asterisk will match all entries. An asterisk is allowed for INCLUDE or EXCLUDE requests, but such a subscription can not be separately deleted or queried. See Usage Note [3](#).

## Usage Notes

1. You may use one or more of the prefix keywords (REQUEST, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Notifications associated with subscriptions established by this command will not be delivered unless an appropriate ASYNCHRONOUS\_UPDATE\_NOTIFICATION\_EXIT. statement is configured in a DirMaint override configuration file. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on the Asynchronous Update Notification exit.
3. Xdata operands (*DATA subscriber\_data\_string*) must be specified in order to query or delete subscriptions which contain subscriber data. For example:

```
DIRM SUBSCRIBE QUERY * * * * DATA *
```

will return all subscriptions, while:

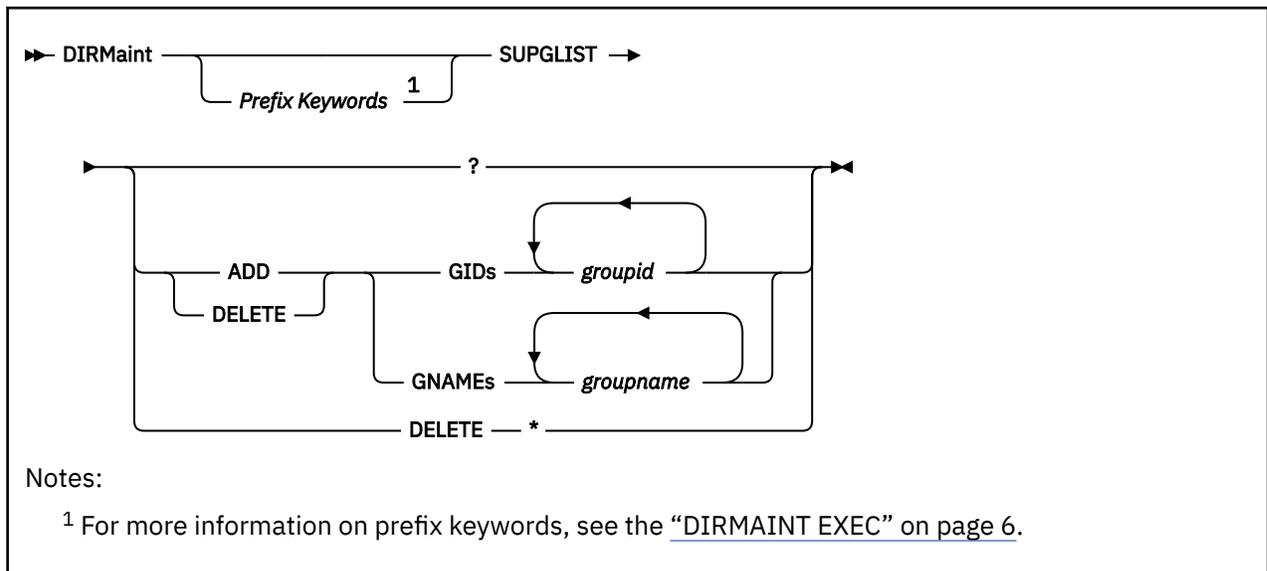
```
DIRM SUBSCRIBE QUERY * * * *
```

will return all subscriptions that have *no* subscriber data.

## Examples

1. DIRM FOR ALL SUBSCRIBE INCLUDE RDR DIRWATCH \* EBCDIC DATA DIRWATCH  
Subscribes the DIRWATCH ID to directory updates for ALL users via RDR file in EBCDIC.
2. DIRM FOR DATAMOVE SUBSCRIBE EXCLUDE RDR DIRWATCH \* EBCDIC DATA DIRWATCH  
Excludes directory updates for the DATAMOVE ID.
3. DIRM FOR \* SUBSCRIBE QUERY RDR DIRWATCH \* \* DATA \*  
Shows which subscriptions the DIRWATCH ID has in effect.

## SUPGLIST



### Purpose

Use the SUPGLIST operand of the DIRMAINT command to add, delete, or query Group IDs or Group Names on the DirMaint supplemental group associated with a profile, user, or identity entry.

### Operands

?

requests that the contents of the current supplemental group list be returned in a message.

**ADD**

**DELETE**

requests that the specified value be added to, or deleted from the current supplemental group list.

**DELETE \***

requests that the entire statement be deleted.

**GIDS *groupid***

specifies that the following values are to be considered group IDs (GIDs).

**GNAMEs *groupname***

specifies that the following values are to be considered group names (GNAMEs).

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOSys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. DirMaint normally represents POSIXGLIST's as a single POSIXGLIST statement with all GIDs preceding GNAMEs. Regardless of their position in the directory entry, they are placed on a single record.

In some environments this may have undesirable effects. The user's supplementary GID list consists of up to 32 unique GIDs. The primary GID (from the POSIXINFO statement) is always part of this list. The remainder of the supplementary list is built by consulting the users POSIXGLIST statements, in order, from the directory entry. If the directory entry has more than 32 items listed, the supplementary list may be built incorrectly due to the method used by DirMaint to represent the POSIXGLIST internally.

By adding a GID or GNAME to the supplemental Glist, DirMaint will generate a POSIXGLIST statement with these GIDs and GNAMEs first, and any remaining GIDs and GNAMEs will follow on a subsequent POSIXGLIST statement. Using this method a user can choose which groups will be specified in a position to be used on their supplemental list.

3. When adding a GID or GNAME, the request is rejected if the value already exists on the list.
4. When deleting a GID or GNAME, the request is rejected if the value is not found on the current list.
5. Adding a value to the supplemental Glist that is not on the actual POSIXGLIST does not result in an actual POSIXGLIST entry. Remember, the supplemental Glist is a DirMaint construct (its internal representation is a comment within the directory) that controls how the actual POSIXGLIST statement is generated in the resultant directory. If a GID or GNAME is present on the supplementary Glist and is also on the actual POSIXGLIST, it is placed in the first generated POSIXGLIST. If an entry is placed on the supplemental Glist that does not exist on the actual POSIXGLIST, no actual POSIXGLIST is generated. For example, if the internal representation of the directory is:

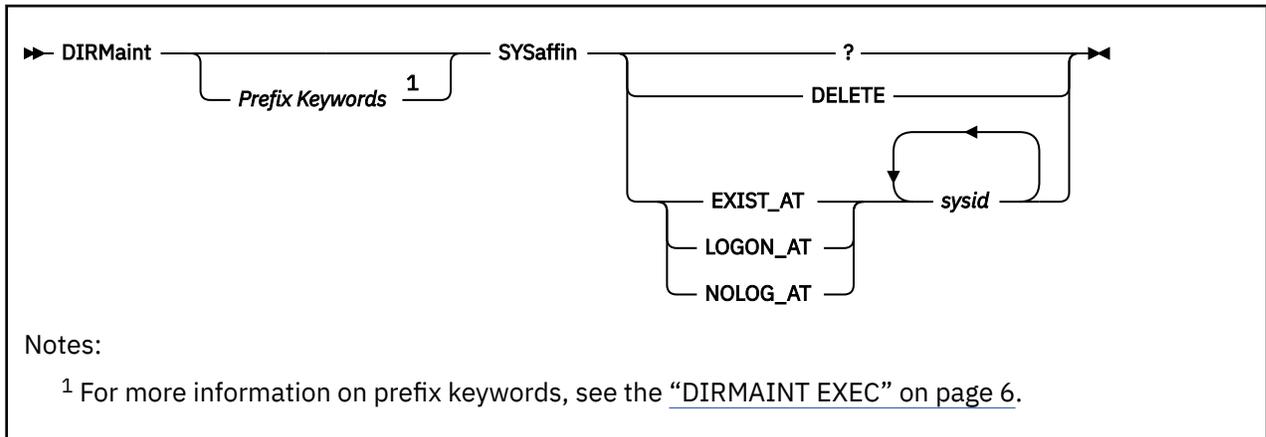
```
-
-
*DVGHLIST= GID 1 GNAME MyG2 MyG3 Unk1 Unk2
POSIXGLIST GID 1 2 3 GNAME MyG1 MyG2 MyG3 MyG4
-
-
```

The resultant directory will look as follows:

```
-
-
*DVGHLIST= GID 1 GNAME MyG2 MyG3 Unk1 Unk2
POSIXGLIST GID 1 GNAME MyG2 MyG3
POSIXGLIST GID 2 3 GNAME MyG1 MyG4
-
-
```

**Note:** All the items that were on the DirMaint supplemental Glist (\*DVGHLIST=) and also on the actual Glist ended up on the first generated Glist, all remaining values were represented on the second Glist. The values from the DirMaint supplemental Glist without a corresponding Glist entry remain as comments.

## SYSAFFIN



### Purpose

Use the SYSAFFIN operand of the DIRMAINT command to support the prefix form of System Affinity.

### Operands

?

requests that the current values associated with the prefix form of System Affinity be returned in a message.

### DELETE

requests that the current prefix System Affinity statement be deleted.

### EXIST\_AT

specifies that the following user definition is to be compiled into the object directory only when running one of the listed systems.

### LOGON\_AT

specifies that the following user definition is to be compiled into the object directory but the associated virtual machine will only operate on the specified systems. It will be NOLOGGED on all other systems.

### NOLOG\_AT

specifies that the following user definition is to be compiled into the object directory but the associated virtual machine will not operate on the specified systems. However, the machine and all its resources are to exist as directory entries in the object directories of all machines.

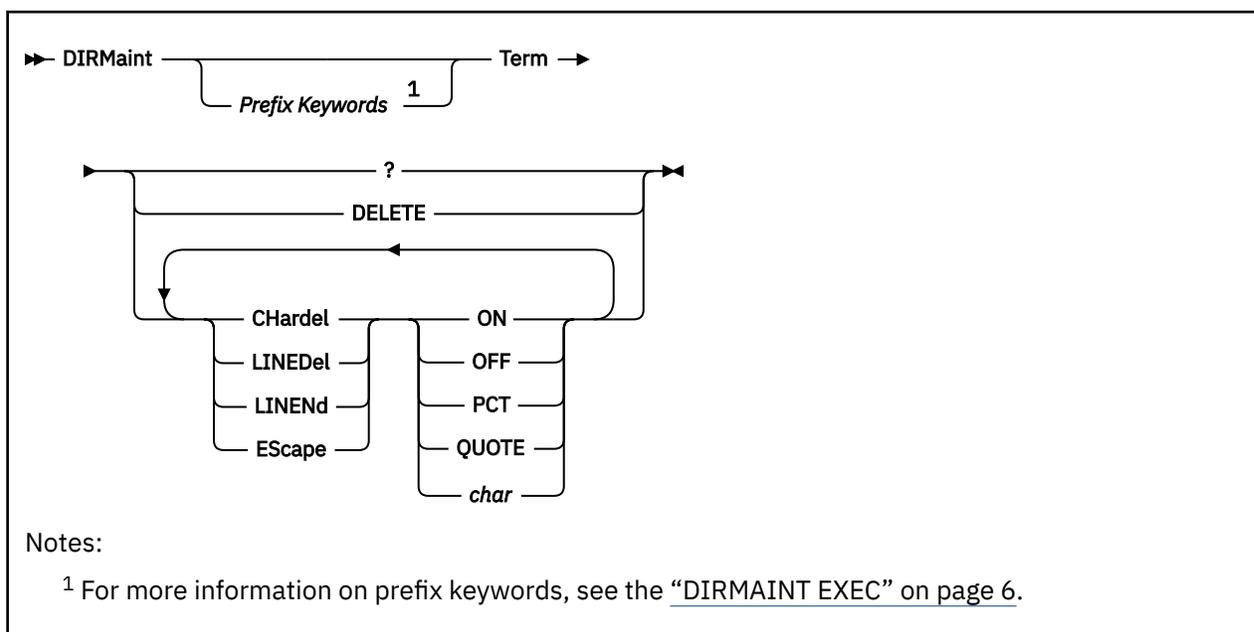
### sysid

a system ID associated with the prefix format of System Affinity.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. For more information on the SYSAFFIN directory statement, see [z/VM: CP Planning and Administration](#).
3. Use the DirMaint ATnode prefix operand to specify the internal form of System Affinity.
4. This operand is only valid on user directory entries.
5. An attempt to use this operand on a directory entry that has the internal format of System Affinity is tolerated. However, the operation is rejected if a conflicting definition results.

## TERM



### Purpose

Use the TERM operand of the DIRMAINT command to allow you to establish customized logical line editing symbols, or modify an existing set of symbols on the USER or IDENTITY control statement. These symbols are then in effect for all terminal sessions.

### Operands

#### CHardel

Indicates that one of the following options applies to the character delete symbol:

##### ON

specifies that the default character delete symbol is @.

##### OFF

specifies that no character delete symbol is to be active.

##### PCT

Indicates that editing character is to be a percent sign (%).

##### QUOTE

Indicates that editing character is to be a single quote (').

##### *char*

indicates that the value specified for *char* is to become the character delete symbol.

#### LINEDel

Indicates that one of the following options applies to the logical line delete symbol:

##### ON

specifies that the default line delete symbol is ␣.

##### OFF

specifies that no logical line delete symbol is to be active.

##### PCT

Indicates that editing character is to be a percent sign (%).

**QUOTE**

Indicates that editing character is to be a single quote (').

**char**

indicates that the value specified for *char* is to become the logical line delete symbol.

**LINEND**

Indicates that one of the following options applies to the logical line end symbol:

**ON**

specifies that the default logical line end symbol is #.

**OFF**

specifies that no logical line end symbol is to be active.

**PCT**

Indicates that editing character is to be a percent sign (%).

**QUOTE**

Indicates that editing character is to be a single quote (').

**char**

indicates that the value specified for *char* is to become the logical line end symbol.

**EScape**

Indicates that one of the following options applies to the logical escape symbol:

**ON**

specifies that the default logical escape symbol is ".

**OFF**

specifies that no logical escape symbol is to be active.

**PCT**

Indicates that editing character is to be a percent sign (%).

**QUOTE**

Indicates that editing character is to be a single quote (').

**char**

indicates that the value specified for *char* is to become the logical escape symbol.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. The CP TERMINAL command allows you to establish logical line editing symbols that will then be in effect for the remainder of your terminal session. These symbols can be used for character deletion, line deletion, line end, or escape character purposes.

For more information on editing symbols, see the CP TERMINAL command in the see [z/VM: CP Commands and Utilities Reference](#).

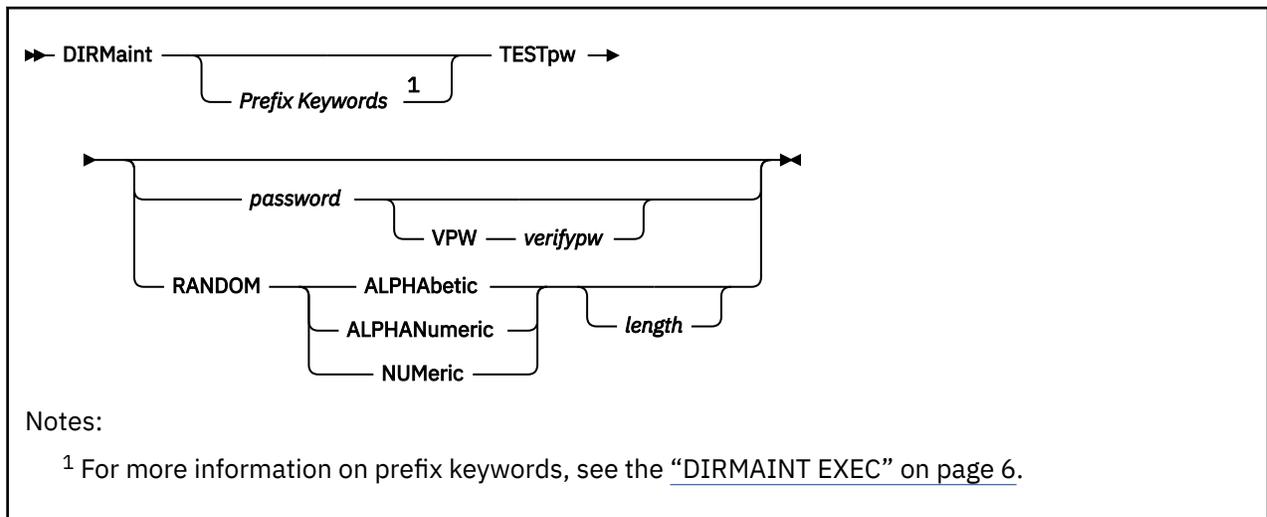
3. You can specify more than one option in a single DIRMAINT command. For example, the following command would turn on the standard character delete symbol @, set the escape character to !, set the line end character to %, and the line delete character to '.

```
DIRM TERM CH ON ES ! LINEND PCT LINEDEL QUOTE
```

**Note:** The PCT is used for the percent symbol (%) and QUOTE is used for the single quotation mark (').

4. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you receive a message informing you that the source and online directories have been updated.

## TESTPW



### Purpose

Use the TESTPW operand of the DIRMAINT command to allow users to check a proposed new CP logon password for acceptability with installation policies. Users are prompted to enter their new password if it was not specified with the TESTPW operand. The password is masked out as blanks on the console as they are typed.

The new password is presented to any locally-defined user exit procedure. If the password is not acceptable, the user exit should inform the user and explain why the password was rejected. The TESTPW operand is for screening only, it will not update the password.

### Operands

#### **password**

specifies the value to be used as a password.

#### **VPW verifypw**

specifies the same password again for typographical verification.

#### **RANDOM**

requests that a random password be generated by calling the PASSWORD\_RANDOM\_GENERATOR\_USER\_EXIT.

#### **ALPHAbetic**

requests that the random generator exit be provided with the keyword ALPHABETIC in the requested algorithm field.

#### **ALPHANumeric**

requests that the random generator exit be provided with the keyword ALPHANUMERIC in the requested algorithm field.

#### **NUMeric**

requests that the random generator exit be provided with the keyword NUMERIC in the requested algorithm field.

#### **length**

requests that the random generator exit be provided with the value *length* as an optional parameter.

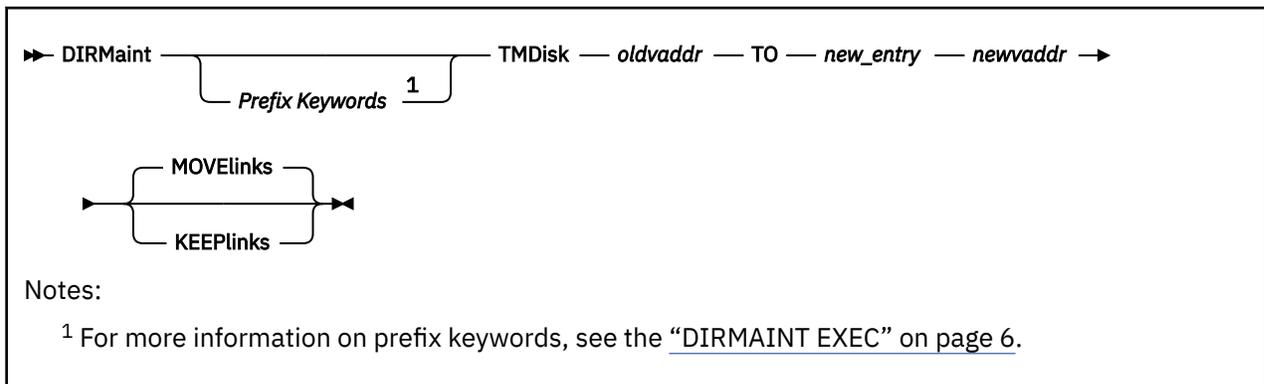
## Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, MENU, PROMPT) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. (Note that PROMPT is a default for this command, and will be assumed unless you override it with MENU.) The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. As part of the verification of passwords, the PASSWORD\_SYNTAX\_CHECKING\_USER\_EXIT routine is called.

For more information, see *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*.

4. Passwords generated by a call to the PASSWORD\_RANDOM\_GENERATOR\_USER\_EXIT are still passed through the PASSWORD\_SYNTAX\_CHECKING\_USER\_EXIT for verification.
5. Your installation may maintain a list of *restricted passwords*, that is, specific passwords whose use are restricted. The transaction will be rejected with an error message if your new password is on the restricted list.
6. All passwords are checked by the installations PASSWORD\_SYNTAX\_CHECKING\_EXIT routine, even though they have already passed the checking done by the USER exit.

## TMDISK



### Purpose

Use the TMDISK operand of the DIRMAINT command to transfer ownership of a minidisk from one user, identity, or subconfiguration entry to another user, identity, or subconfiguration entry.

### Operands

#### **oldvaddr**

is the virtual device address of the minidisk in the current user's configuration.

#### **TO**

is a required keyword.

#### **new\_entry**

is the name of the directory entry to be given the minidisk.

#### **newvaddr**

is the virtual device address of the minidisk when placed in the new user's configuration.

#### **MOVElinks**

requests that all links to the source user's minidisk be deleted and a link to the new user's minidisk be generated in their place.

#### **KEEPlinks**

requests that all links be left as they are currently defined.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. This operand should be used when the minidisk and all files on it are to be transferred to a different user ID.
3. The new owner will inherit all files currently on the minidisk. The minidisk in the new owner's configuration will have the same access mode and access passwords as established by the original owner.
4. If the intent is merely to transfer the space, then the original owner should be told that they should initialize or reformat the disk before the transfer is done. This may merely be a matter of individual privacy, or may involve company policy or security. Individual installations may emphasize this procedure as appropriate.
5. Any links to the transferred minidisk are changed to link to the new owner's minidisk if the MOVELINKS option is used. This is done as low priority work element (batch). If the *linker* issues a LINK DELETE

## TMDISK

request before the low priority command executes, the low priority command to change the link to the new owner will fail. This *error* condition can be safely ignored.

6. If the AT prefix is specified for a subconfiguration entry when the TMDISK operand is issued, the AT keyword value will be used to select the DATAMOVE machine configured at the specified node. In this respect, the FORuser prefix keyword must specify the subconfiguration entry name in order to modify a subconfiguration entry with the TMDISK operand.
7. If your system specifically has RACF installed as the ESM, see the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for configuring DirMaint's RACF connector support to provide automatic communication with RACF.

## UNDOSSI

►► DIRMaint ——— UNDOSSI — nodeID ◄◄  
 └── Prefix Keywords 1 ─┘

Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

Use the UNDOSSI operand of the DIRMAINT command to roll back BUILD statement changes done by the SSI operand and to remove the SSI operand from the DIRECTORY statement.

### Operands

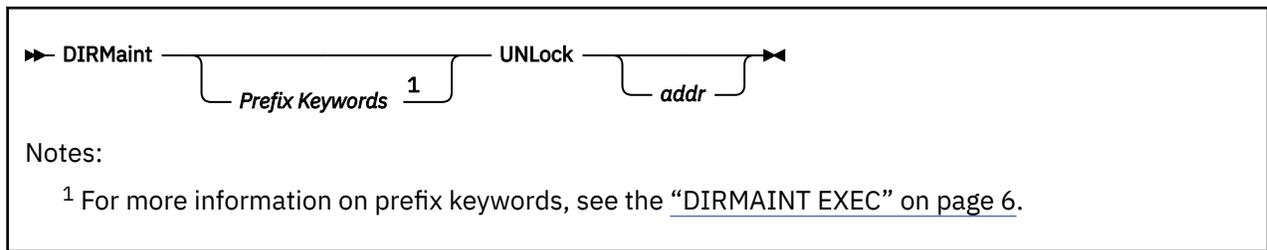
#### *nodeID*

specifies the node in an SSI cluster, as used in BUILD statements.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords do not apply to this command. They are accepted without error, but otherwise ignored.
2. Only a directory with one system in an SSI cluster can be changed with this operand. If there is more than one BUILD per identity, this operand will fail. This operand will also fail if BUILD statements exist containing a different node ID than the node ID specified.
3. You can use the DIRECTORY operand to add options to a DIRECTORY statement. See [“DIRECTORY”](#) on page 93.

## UNLOCK



### Purpose

Use the UNLOCK operand of the DIRMAINT command to remove a lock against a directory entry or remove a lock against a specific minidisk within a directory entry.

### Operands

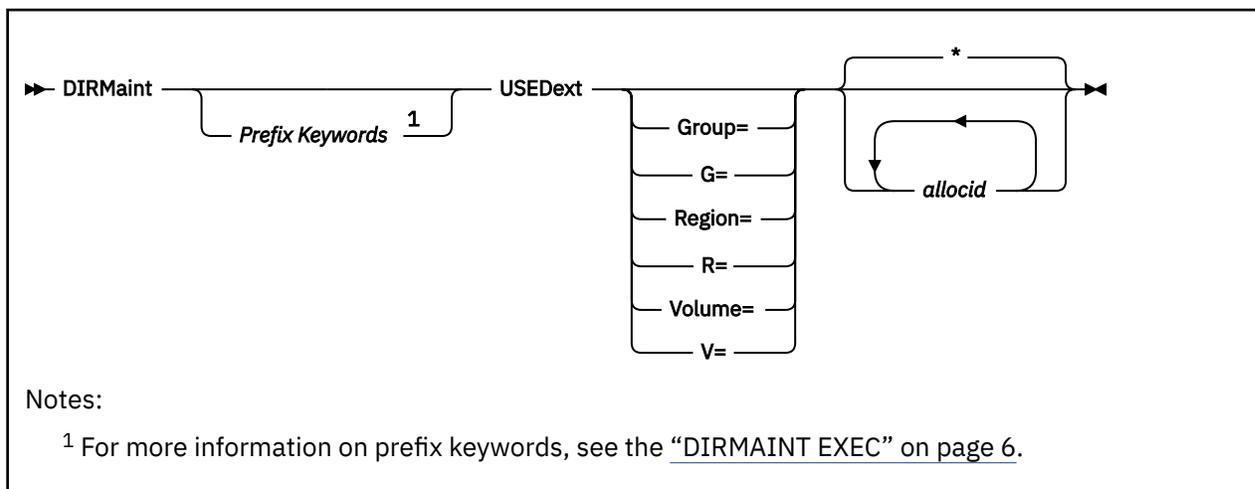
#### *addr*

is the virtual address of the minidisk to be unlocked.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. They are accepted without error, but otherwise ignored.
2. There are two circumstances that lead to the setting of a directory entry lock:
  - a. Use of the GET operand (without the NOLOCK option)
  - b. Explicit use of the LOCK function.
3. The UNLOCK operand can be used whenever it is desirable to allow a user to resume making changes directly to their own directory entries.
4. The UNLOCK operand will be invoked automatically whenever the REPLACE operand is used.
5. The *addr* option provides support to unlock a minidisk. Device locks are set by REPLACE, AMDISK, CMDISK, and DMDISK operands.

## USEDEXT



### Purpose

Use the USEDEXT operand of the DIRMAINT command to generate a listing of allocated extents on volumes, groups, and regions used to contain minidisks.

### Operands

#### *allocid*

specifies the search domain. If VOLUME= is specified, then *allocid* is the volume ID. If GROUP= is specified, then *allocid* is the group name. If REGION= is specified, then *allocid* is the region name.

If none are specified, then volume ID is assumed, but the reports will be returned in the format used before DirMaint function level 510. This older format is not intended to be used as a programming interface. See “Reports” on page 282 for the newer formats.

#### **volume ID**

is the volume label of the volume(s) to be scanned, with each specific volume ID being 1 to 6 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +.

If a 1 to 6 character volume label is specified, the listing will contain only allocated extents on that volume.

If a partially-qualified volume label of 1 to 5 characters followed by an asterisk (\*) is specified, the listing will include allocated extents on all volumes beginning with the qualified volume ID. For example, if the total set of volumes containing minidisks are:

```
ABC
ABC12
ABC456
ABX445
VOL001
VSZ777
```

then specifying volume ID as ABC\* will cause volumes ABC, ABC12 and ABC456 to be scanned. Specifying ABC would qualify only the discrete volume ABC. Specifying V\* would cause volumes VOL001 and VSZ777 to be qualified.

If the volume ID is omitted, or if an \* is specified, then all volumes used to house minidisks will be included in the output.

Multiple discrete volumes (or multiple partially-qualified volumes) are also allowed.

**group name**

is the name specifying the group(s) to be scanned, with each specific group name being 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +. The same rules apply for partially-qualified group names as for volume IDs (above). Note that specifying group names on this command will result in a volume space report for each of those volumes containing regions that are part of those groups.

**region name**

is the name specifying the region(s) to be scanned, with each specific region name being 1 to 8 characters from the following list: A-Z, 0-9, #, @, \$, :, - and +. The same rules apply for partially-qualified region names as for volume IDs (above). Note that specifying region names on this command will result in a volume space report for each of those volumes containing those regions.

**Reports**

Reports returned by the USEDext command, with the G=, GROUP=, R=, REGION=, V=, or VOLUME= keyword specified, appear in the following format:

USEDEXT *operands*

GROUP	REGION	VOLUME	START	SIZE	(END)	OWNER	ADDR	SA	NOTES
gggggggg	rrrrrrrr	vvvvvvv	nnnnnnnnnn	nnnnnnnnnn	nnnnnnnnnn	uuuuuuuu	aaaa	ssssssss	notes
gggggggg	rrrrrrrr	vvvvvvv	nnnnnnnnnn	nnnnnnnnnn	nnnnnnnnnn	uuuuuuuu	aaaa	ssssssss	notes

Unless the RETVIA/RETURNVIA prefix keyword is used, the result will be a reader file in the above format. There will be one header line per request. There will usually be one detail line per minidisk entry. The "GROUP" column will show an "\*" for a REGION or VOLUME report. The "REGION" column will show an "\*" for a VOLUME report. The (END) column is calculated as (START+SIZE)-1. If the minidisk is contained within multiple overlapping regions, or in a region that is included in more than one group, there may be multiple entries for that minidisk record.

If DirMaint is running in a single system image (SSI) cluster, reports returned by the USEDEXT appear in the following format:

USEDEXT *operands*

GROUP	REGION	VOLUME	START	SIZE	(END)	OWNER	ADDR	SSI NODE	NOTES
gggggggg	rrrrrrrr	vvvvvvv	nnnnnnnnnn	nnnnnnnnnn	nnnnnnnnnn	uuuuuuuu	aaaa	xxxxxxxx	notes
gggggggg	rrrrrrrr	vvvvvvv	nnnnnnnnnn	nnnnnnnnnn	nnnnnnnnnn	uuuuuuuu	aaaa	xxxxxxxx	notes

where the SSI NODE column shows the SSI cluster member to which the subconfiguration entry associated with the owning identity (OWNER) belongs.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, MENU, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. Optional parameters are allowed on this command, but none are required. Use the MENU prefix operand to display the menu for this command so you can select or fill in the optional parameters.
3. The name of the file returned to the invoker of this command is dependant on the method used to invoke the command.

**allocid USEDEXT**

specifies the invocation was made with a specific group/region/volume name specified.

**alloc\$ USEDEXT**

specifies the invocation was made with a partially qualified group/region/volume specified.

**ALLGRPS USEDEXT**

specifies the invocation was made with a \* as the qualifier for group.

**ALLREGS USEDEXT**

specifies the invocation was made with a \* as the qualifier for region.

**ALLVOLS USEDEXT**

specifies the invocation was made with a \* as the qualifier for volume.

**MULTIGRP USEDEXT**

specifies the invocation was made with more than one qualifier for group.

**MULTIREG USEDEXT**

specifies the invocation was made with more than one qualifier for region.

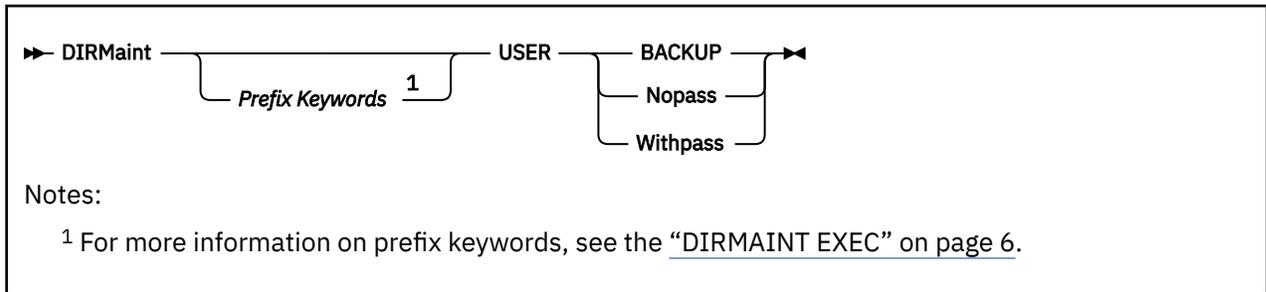
**MULTIVOL USEDEXT**

specifies the invocation was made with more than one qualifier for volume.

**Note:** When there are no free extents or used extents meeting the search criteria the corresponding file returned will contain only a header with no data.

4. Processing consists of making one pass over each volume control file (one per volume ID) and listing all MDISK control statements found that are allocated on volumes within the search domain defined by *allocid*. As each disk in the candidate list is added to the report, its starting cylinder or block number is sorted.
5. In the normal case, the starting cylinder or block for one minidisk will be greater than the ending cylinder or block for the preceding minidisk on the volume. If the starting cylinder or block for any minidisk is less than or equal to the ending cylinder or block of the preceding minidisk, an exception is flagged. If the two minidisks have the:
  - Same starting and ending location, virtual disk address, and owner, differing only in the system affinity associated with the two entries (with neither system affinity being an asterisk), then the condition is flagged as shared.
  - Same starting and ending location, address, owner, differing only in the system affinity with one of them being an asterisk or if the system affinity values are also the same, then the condition is flagged as an error.
  - Different starting locations, or different ending locations, or different virtual addresses, or different owners, then the condition is flagged as an overlap. If any exceptions are flagged, a DIAGNOSE file will be generated in addition to the requested report file.
6. The "Special User IDs in the Default User Directory (USER DIRECT)" section in *z/VM: Installation Guide* recommends using and/or updating special dummy user IDs in the default user directory (USER DIRECT) to track allocation of CP areas on a volume. For example, the \$DIRECT\$ user ID tracks directory cylinders/blocks. The CP QUERY ALLOC MAP command identifies CP reserved areas, and these areas are tracked by DirMaint in the command output as .DRCT., .PAGE., .SPOOL., and .TDISK. areas. The USEDEXT command does not list overlaps between a special user ID and the allocation extents information when the extents exactly match. This avoids an OVERLAP warning on those extents.

## USER



### Purpose

Use the USER operand of the DIRMAINT command to obtain a copy of the source directory in a single file format. It is also used to generate the USER BACKUP file, which is the primary disk backup of the source directory.

### Operands

#### BACKUP

specifies that a file is not returned to the invoker, but a composite backup file named USER BACKUP is generated on the backup disks. The previous USER BACKUP file is retained and renamed to USER BACKOLD. This feature requires that the backup disk be large enough to contain two copies.

**Note:** The difference between the BACKUP and the USER BACKUP operands is that the USER BACKUP operand causes the creation of the USER BACKUP file only. The BACKUP operand invokes the full backup procedure which also backs up the auxiliary control files.

#### Nopass

sends an edited copy of the source directory to your virtual card reader named USER NOPASS. All logon passwords are replaced with XXXXXXXX. All minidisk access passwords are replaced (if they exist) as the:

- Read password is replaced with READPASS
- Write password is replaced with WRITPASS
- Multiple write password is replaced with MULTPASS.

Any APPCPASS statements in the source directory will have the password field set to XXXXXXXX.

USER and IDENTITY statements that have a value of the following keywords in the password field will show the keyword instead of masking it:

- AUTOONLY
- LBYONLY
- NOLOG
- NOPASS

#### Withpass

sends an unmodified copy of the source directory to your virtual card reader named USER WITHPASS.

**Note:** This file contains all passwords for all users at your installation. Take the appropriate security precautions.

### Usage Notes

1. You may use one or more of the prefix keywords (REQest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only

be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.

2. The NOPASS option produces an edited version of the source directory that can be printed and kept available for reference. While passwords are not revealed, other items of information may be considered sensitive at your installation. Therefore, you should take appropriate security measures with the printed listing.
3. The USER BACKUP command may be used by authorized users or external processes that require read-only access to the current state of the directory source. After the command completes, staff members may use the DIRM SEND command to request that the newly generated USER BACKUP file be sent to their virtual reader, or they can access the DirMaint 1DB and process the file from there.
4. For the NOPASS and WITHPASS options, the file sent to your virtual reader will be in single file format. It should be read on to your disk using the CMS RECEIVE command.
5. When creating a new source directory from either a USER WITHPASS or USER BACKUP, the previous USER DIRECT file must be erased before executing the DVHBEGIN EXEC otherwise the new USER INPUT file will not be used.

## USERMAP



### Notes:

<sup>1</sup> For more information on prefix keywords, see the [“DIRMAINT EXEC”](#) on page 6.

### Purpose

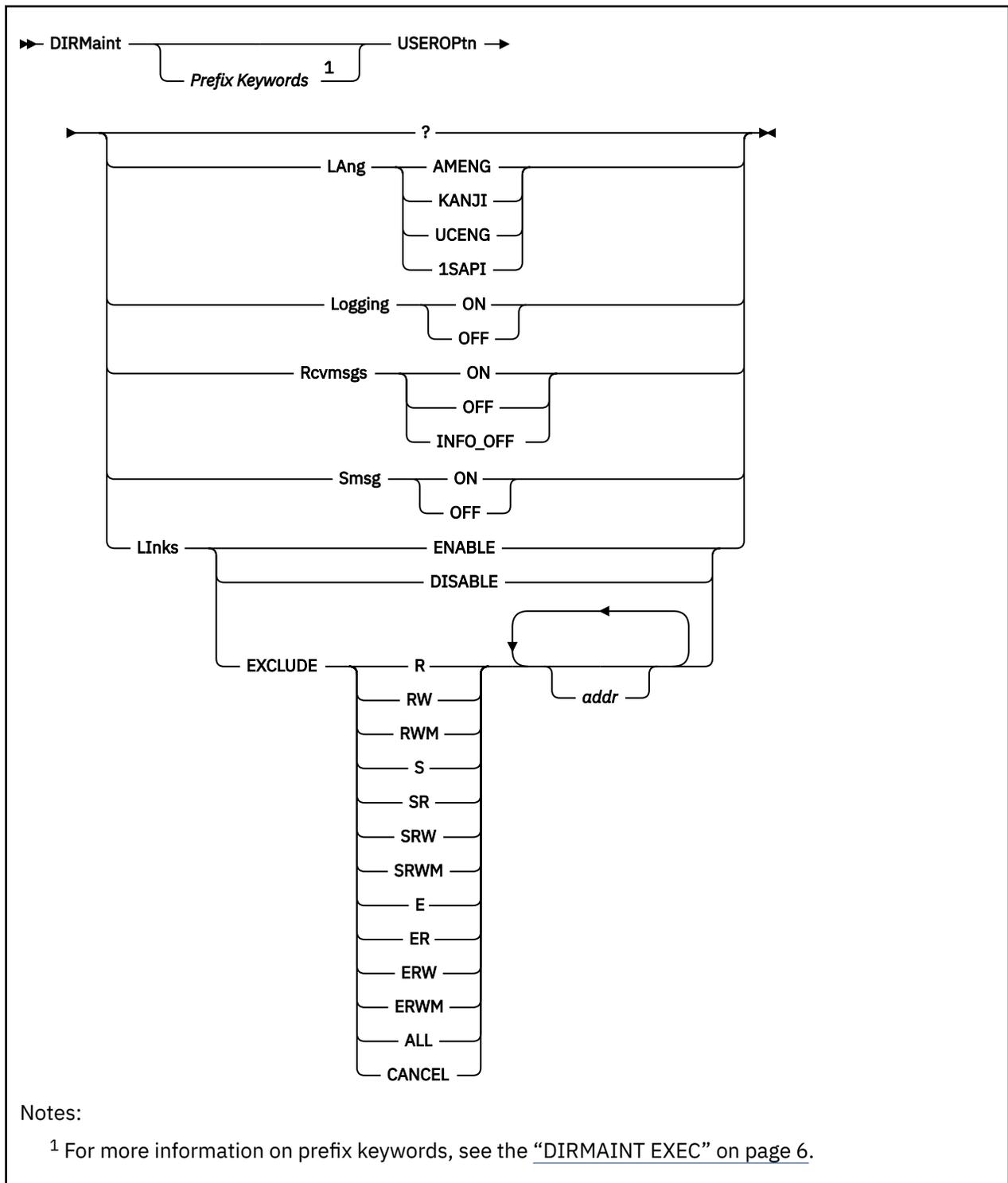
Use the USERMAP operand of the DIRMAINT command to generate a list of profile, user, identity, and subconfiguration entries currently in the source directory. The contents of the file that DirMaint sends is called USER DIRECT. This file is the same as a CP LOAD Control Statement.

For more information on the CP LOAD Control statement, see [z/VM: CP Planning and Administration](#).

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, RETurnvia/RETVia) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.
2. This operand generates a file named USER DIRECT containing all profile, user, identity, and subconfiguration entry names and their starting record numbers within the source directory, and sends it to your virtual card reader. The file can be loaded to your minidisk using the CMS RECEIVE command.

# USEROPTN



## Purpose

Use the USEROPTN operand of the DIRMAINT command to control the interaction between your virtual machine and the DIRMAINT virtual machine.

## Operands

?

requests that the current settings be returned in a message. You will receive back the language specified with the LAng parameter, as well as a set of numerical values for the other parameters, as follows:

```
Links 0 = DISABLED, 3 = ENABLED
Logging 1 = ON, 2 = OFF
Rcvmsgs 0 = OFF, 1 = ON, 2 = INFO OFF
Smsg 0 = OFF, 1 = ON
NeedPass 0 = NO, 1 = YES
Lang AMENG
```

**LAng langid**

specifies the default DirMaint language associated with this entry.

**Logging**

specifies whether the messages sent to your user ID by the DIRMAINT virtual machine are to be logged if they cannot be sent to you. The file name of the log file will be the user ID name, with a file type of the node ID from where the report was created.

**Note:** The USER LOG identifies the *userid* of the issuer of the command issuer and uses a file type of *nodeid* of that user.

**ON**

specifies messages are to be logged. This is the IBM supplied default. The default may be different on your system.

**OFF**

indicates that those messages will be lost.

**Rcvmsgs**

specifies whether your virtual machine is accepting messages from the DIRMAINT virtual machine.

**ON**

indicates that messages from the DIRMAINT virtual machine be sent to your terminal. This is the IBM supplied default. The default may be different on your system. If you are not currently receiving messages, but the LOGGING=ON option is in effect, these messages will be logged.

**OFF**

indicates that messages from the DIRMAINT virtual machine will not be sent to your terminal. OFF automatically invokes the LOGGING=ON option.

**Note:** The USEROPTN RCVMSG OFF will not automatically issue the LOGGING=ON option.

**INFO\_OFF**

indicates that informational messages from the DIRMAINT virtual machine will not be sent to your terminal, but will be logged if the LOGGING=ON option is in effect.

**Smsg**

specifies how messages are to be sent to your virtual machine.

**ON**

indicates sending messages as virtual storage images by means of the CP SMSG facility.

**OFF**

indicates sending messages in visual form by means of the CP MSG facility. This is the IBM supplied default. The default may be different on your system.

**LInks**

specifies whether other general users can establish links to your minidisks using the DIRM LINK command.

**ENABLED**

allows other general users to establish links to minidisks owned by your user ID.

**DISABLED**

prevents other general users from establishing links to minidisks owned by your user ID. This is the IBM supplied default. The default may be different on your system.

**EXCLUDE *mode addr***

specifies that any entry for the indicated minidisk be added to, or the existing entry be changed or removed from, the DVHLINK EXCLUDE control file. A single entry for the *addr* minidisk in the DVHLINK EXCLUDE file prevents multiple entries for *mode* links from appearing in the DVHLINKS FILE.

Valid *mode* requests include:

**R**

Read links (R and RR) should be excluded.

**RW**

Read and Write links (R, RR, W and WR) should be excluded.

**RWM**

Read, Write and Multi Write links (all except S and E links) should be excluded.

**S**

Stable links (any link using the S suffix) should be excluded.

**SR**

Stable Read links (any read link using the S suffix) should be excluded.

**SRW**

Stable Read and Write links (any read or write link using the S suffix) should be excluded.

**SRWM**

Stable Read, Write and Multi Write links (all except Exclusive links) should be excluded.

**E**

Exclusive or Stable links (any link using the S or E suffix) should be excluded.

**ER**

Exclusive or Stable Read links (any read link using the S or E suffix) should be excluded.

**ERW**

Exclusive or Stable Read and Write links (any read or write link using the S or E suffix) should be excluded.

**ERWM**

Exclusive or Stable Read, Write and Multi Write links (all links) should be excluded.

**ALL**

All links should be excluded.

**CANCEL**

The entry should be removed from the link exclude control file.

An excluded link is not included in DirMaint's link inventory control file. Therefore, links to excluded devices are not:

- Shown in the result of a REVIEW command.
- Changed by a CHNGID or CHVADDR command.
- Moved by a TMDISK command (MOVELINKS option).
- Deleted by a DMDISK command (MOVELINKS option).
- Deleted by a DLINK \* command.

To have these commands affect an excluded disk, use the

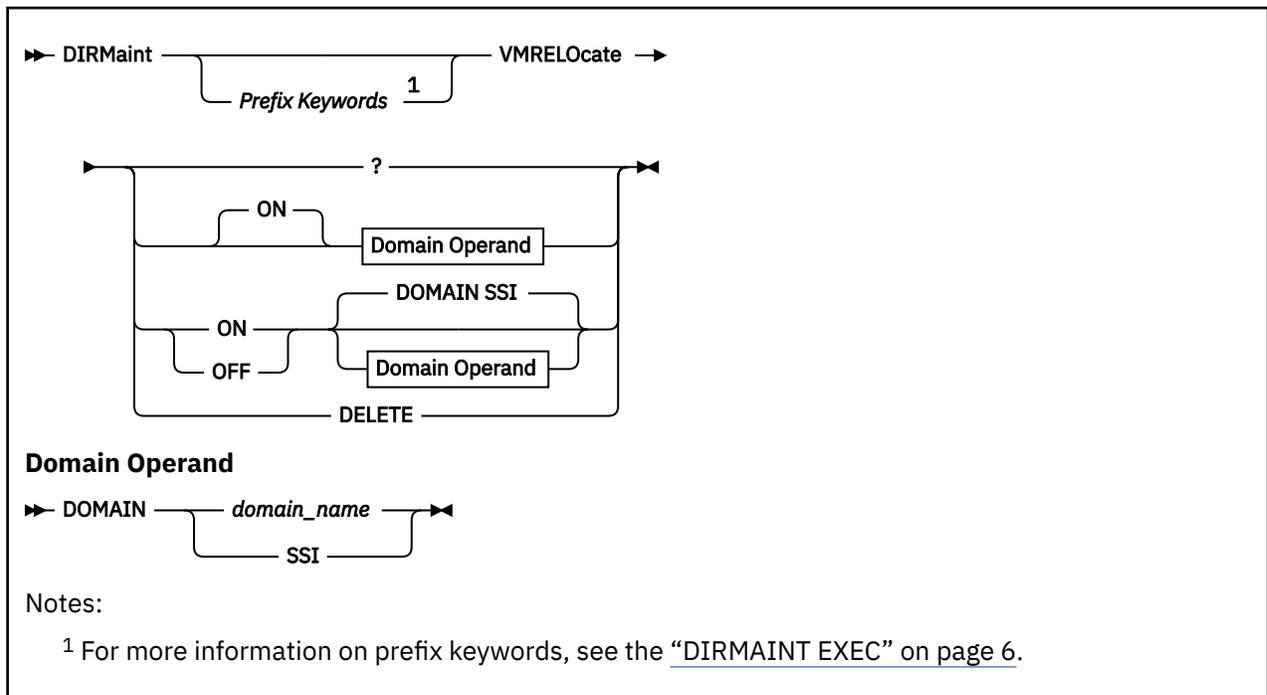
```
USEROPTN LINKS EXCLUDE CANCEL
```

command to remove the entry from the exclude control file, then run RLDEXTN with the ALL option to rebuild the link inventory control file.

**Usage Notes**

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The AT prefix keyword is not applicable to the command. It is accepted without error, but otherwise ignored.
2. A listing of the USEROPTN options currently in effect for your user ID is also returned as part of the output from the DIRM REVIEW operand.
3. Options not specified are not changed.
4. Specifying LINKS DISABLE for an individual user will not prevent an administrative staff member from issuing a DIRM LINK command to that user's minidisk. The protection is only against other general users.
5. The DIRMAINT LINKS ENABLE/DISABLE setting has no effect on CP. Issuing the command DIRM USEROPTN LINKS DISABLE does not inhibit the CP LINK command to your minidisk. It is effective only against DirMaint link requests.
6. Use the RLDEXTN command with the ALL option to rebuild the link file after using USEROPTN to alter the links exclude file or wait for nightly automatic backup processing to take place.
7. To change the NeedPass setting, see [“NEEDPASS” on page 171](#).

## VMRELOCATE



### Purpose

Use the VMRELOCATE operand to query, update, or delete the relocation capability associated with a user or profile entry.

### Operands

?

requests that the current relocation domain setting be returned in a message.

**ON**

enables relocation for the specified user. When user first logs on, relocation will be enabled. If neither ON nor OFF is specified, ON is the default.

**OFF**

disables relocation for the specified user. When the user first logs on, relocation will be disabled.

**DOMAIN *domain\_name***

defines the domain associated with a user when that user first logs on. If DOMAIN is not specified, then the SSI domain is assumed. If a new domain is specified, the user is assigned a virtual architecture level that is the maximum common subset of all the members of the domain.

**DOMAIN SSI**

specifies that the entire SSI membership is associated with the user as a relocation domain. This is the default domain when the user logs on.

**DELETE**

deletes the VMRELOCATE statement from the directory entry. Relocation will be disabled for this user.

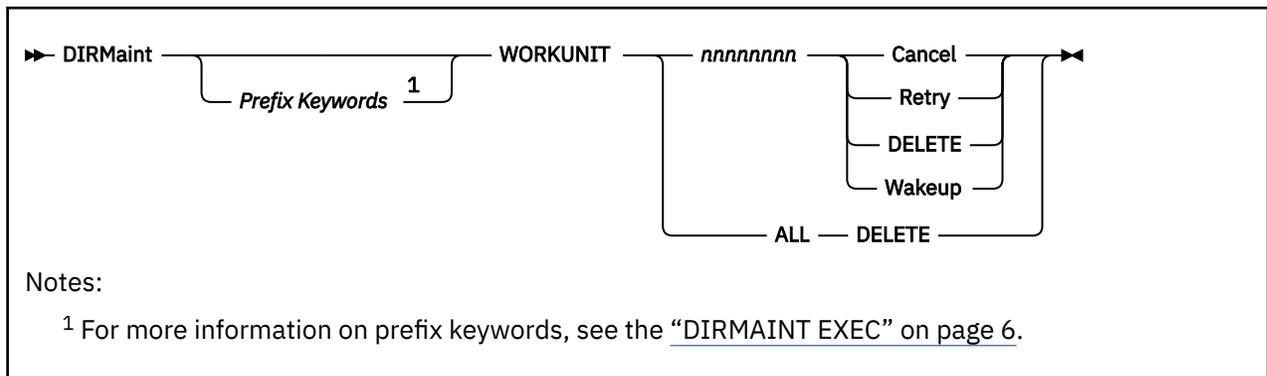
### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.

## VMRELOCATE

2. VMRELOCATE statements are ignored when used in a non-SSI system.
3. VMRELOCATE statements are not allowed in an identity or subconfiguration entry.
4. VMRELOCATE statements are ignored in any profile entry that is included in an identity entry.
5. For more information on the VMRELOCATE directory statement, see [z/VM: CP Planning and Administration](#).

## WORKUNIT



### Purpose

Use the WORKUNIT operand of the DIRMAINT command to manipulate active and failed work units.

### Operands

#### *nnnnnnnn*

is the identification number of the work unit.

#### Cancel

cancels the specified active work unit. This operand sends a cancel transaction to the DATAMOVE machine responsible for this work unit if the work unit is assigned. The cancel completes when the DATAMOVE machine responds. If the work unit is not assigned it is removed from the unassigned queue.

#### Wakeup

requests that a WAKEUP event be sent to the owner of the work unit (or all DATAMOVES if this work unit is unassigned). This operand will cause DATAMOVE to examine its retry queue for any tasks that can be worked on. This is done automatically by DirMaint each time a DIRECT occurs.

#### DELETE

#### ALL DELETE

deletes the specified failed work unit file or *ALL* failed work unit files.

**Note:** The failed work unit file contains information that details how the failure occurred and what steps, if any, were taken by DirMaint to roll back the transaction.

#### Retry

requests that the DATAMOVE control command of an active work unit be retried if it is currently in status ACTIVE or RETRY.

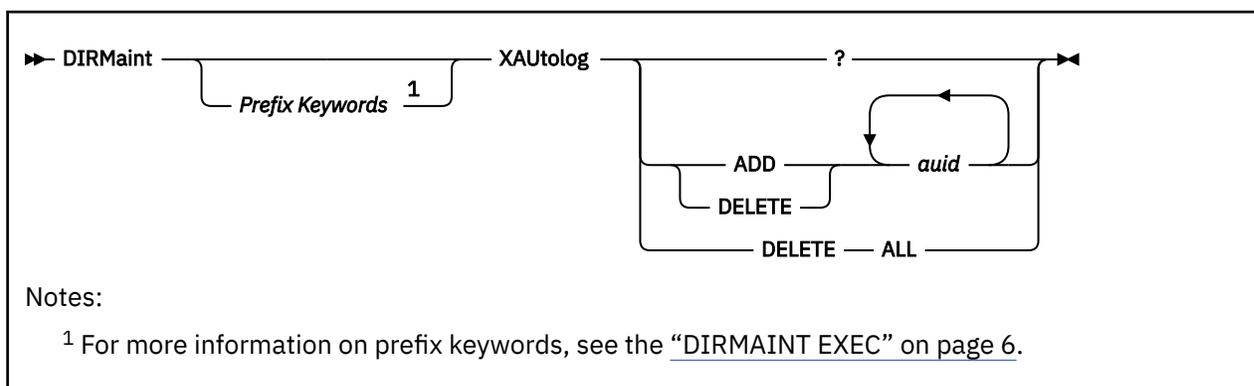
**Note:** This operand redrives the DATAMOVE machine with the DMVCTL request. This operand may be useful if you suspect the DATAMOVE machine or some outside force has *purged the request* from the DATAMOVE transaction queue. If the status of the DMVCTL command is RETRY, this may indicate that the DATAMOVE machine is simply waiting for a DIRECT to occur or for all links to the target device to be released. If this is the case, you should use the WAKEUP operand to force DATAMOVE to reexamine its retry queues.

### Usage Notes

1. You may use one or more of the prefix keywords (REQest, TOsys, ASuser, BYuser, PRESET) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once. The FOR and AT prefix keywords are not applicable to the command. They are accepted without error, but otherwise ignored.

2. A work unit is assigned and built to contain the sequence of commands and user actions required to execute certain complex commands. This mechanism is necessary because operands such as CMDISK could not be completed by a single command processor. In fact, they actually expand into a coordinated sequence of discrete events and are handled asynchronously. For example, the CMDISK operand expands to include allocation of a new extent, copying of data from the old minidisk to the new, confirmation of a successful copy, reformatting of the old minidisk, and release of the old extent.
3. A cancel or retry on a non-existent work unit is rejected with a message.
4. The CANCEL operand results in a CANCEL request being sent to the appropriate DATAMOVE machine (if the work unit is assigned). The DATAMOVE machine will respond with an appropriate CANCEL acknowledgement after it has cleaned up after the work element.  
  
If the work element is not assigned (that is, the request had been accepted but no work has yet been done on it), it is simply deallocated.
5. The WAKEUP operand results in a WAKEUP request being sent to the appropriate DATAMOVE machine (if the work unit is assigned). The DATAMOVE machine will check its retry queue for work that can be done.  
  
If the work unit is not currently assigned, a general WAKEUP request will be sent to all DATAMOVE machines not currently working on a task.
6. The RETRY operand results in the DATAMOVE control command being reissued if it was found to be in RETRY or ACTIVE status. If the DATAMOVE control command is not in one of these two states the command is rejected.
7. See the *z/VM: Directory Maintenance Facility Tailoring and Administration Guide* for more information on how to diagnose and process work units which are in a stalled or retry state.

## XAUTOLOG



### Purpose

Use the XAUTOLOG operand of the DIRMAINT command to add or delete a virtual machine name from an existing AUTOLOG or XAUTOLOG statement for a specified profile, user, or identity entry.

### Operands

?

specifies that the existing AUTOLOG or XAUTOLOG statement is to be queried.

#### ADD

specifies that a user ID (*auid*) is to be added to an existing AUTOLOG or XAUTOLOG statement or a new statement is to be created.

#### DELETE

#### DELETE ALL

specifies that a user ID (*auid*) is to be deleted from the AUTOLOG or XAUTOLOG statement. If the ALL keyword is used the entire statement is deleted.

#### *auid*

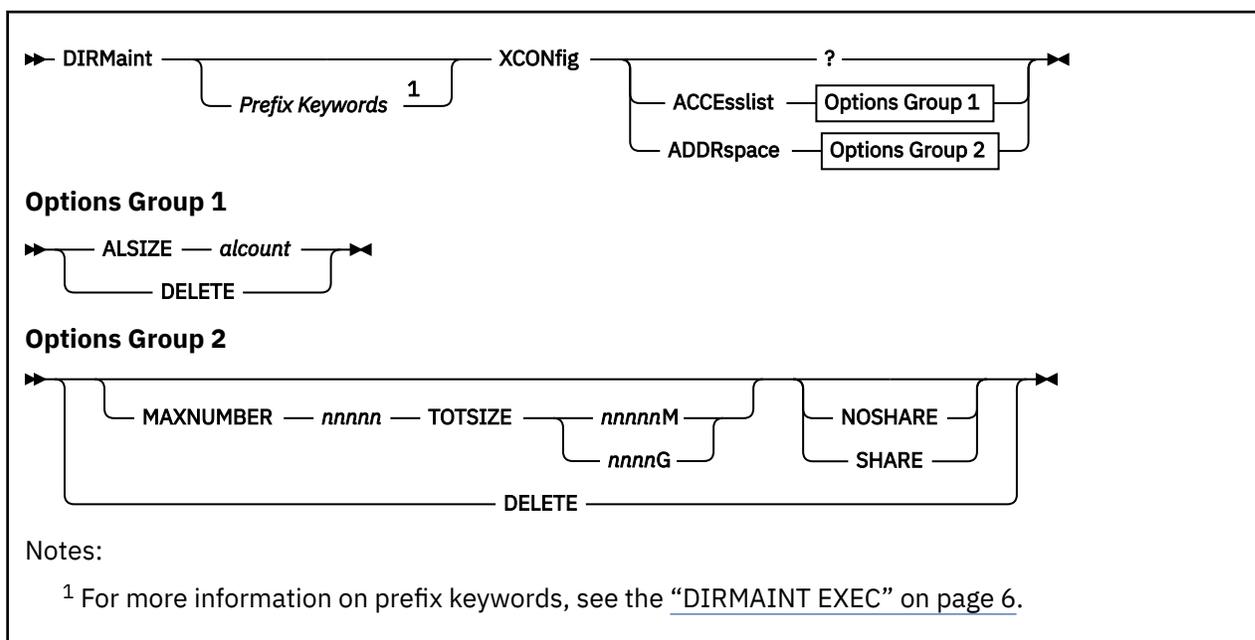
specifies a 1 to 8 character name of the virtual machine authorized to AUTOLOG or XAUTOLOG the specified user ID.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRctxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the AUTOLOG directory statement, see *z/VM: CP Planning and Administration*.
3. XAUTOLOG and AUTOLOG are synonymous but XAUTOLOG is the preferred directory keyword. Therefore, if the keyword AUTOLOG is encountered it is translated to XAUTOLOG.
4. A "DIRM XAUTOLOG ADD *auid*" may result in the creation of an XAUTOLOG statement, if there are no existing XAUTOLOG statements in the directory entry, or no more room on an existing XAUTOLOG statement. Similarly, a "DIRM XAUTOLOG DELETE *auid*" may result in the deletion of an XAUTOLOG statement if there are no more *auids* on the existing XAUTOLOG statement.
5. Specifying DELETE for a user ID not on an AUTOLOG or XAUTOLOG statement will generate an error message. Specifying ADD for a user ID already on an AUTOLOG or XAUTOLOG statement will also generate an error message.
6. A maximum of eight user IDs may be specified through AUTOLOG or XAUTOLOG.

7. The directory change may be brought online immediately by the CP update-in-place facility (DIAGNOSE code X'84'). When this facility is used, you are informed by a message that both the source and online directories have been updated.

## XCONFIG



## Purpose

Use the XCONFIG operand of the DIRMAINT command to allow authorized users to manipulate the XCONFIG directory statement in any directory entry type. The XCONFIG statement specifies control parameters for the extended-configuration facilities provided in the XC virtual machine architecture.

## Operands

?

requests that the current settings of the XCONFIG statements be returned in a message.

### ACCEslist

specifies the size of the host access list to be provided for this virtual machine.

### ADDRspace

specifies the maximum number of nonprimary address spaces and the total size in bytes of all nonprimary address spaces that the virtual machine can own simultaneously.

### ALSIZE *alcount*

specifies the size, in number of access-list entries, for this virtual machine's host access list.

### MAXNUMBER *nnnnn*

specifies the maximum number of nonprimary address spaces that this virtual machine can create and have existing concurrently.

### TOTSIZe *nnnnnM*

### TOTSIZe *nnnnG*

specifies the maximum total size, in bytes, of all address spaces that this virtual machine can create and have existing concurrently.

### SHARE

### NOSHARE

specifies whether the virtual machine can or cannot use the PERMIT function of the ADRSPACE macroinstruction to make its address spaces available for access by other virtual machines.

### DELETE

requests that the specified statement be deleted from the directory entry.

### Usage Notes

1. You may use one or more of the prefix keywords (REQuest, TOsys, ASuser, BYuser, PRESET, FORuser, ATnode, NODIRectxa) with this command. These prefix keywords may be specified in any order, but each keyword may only be specified once.
2. For more information on the XCONFIG directory statement, see [z/VM: CP Planning and Administration](#).

## Appendix A. DirMaint Command Sets

This table provides a summary of the DirMaint commands, the prefix keywords that are honored for each command (keywords not honored are accepted but ignored), whether the command may be used in a BATCH file or through DVHSAPI, whether the command has an available menu or prompt, whether or not password verification is required by default, and the default command sets to which the command is assigned for command levels 150A and 140A.

There are two files, 140CMDS DATADVH and 150CMDS DATADVH, that define which command belongs to what command set(s), and whether or not password verification is required for that command in that command set. The command set definitions can be changed by your installation by editing these files and following the procedures documented in the files themselves. The table shows the IBM supplied command set definitions, as they would be after initial installation and before any tailoring is done.

The IBM default command sets for DirMaint are:

- A** Administration, non-DASD related
- D** DASD Management
- G** General users
- H** Helpdesk
- M** Password Monitor
- O** System Operator
- P** Programmed application commands (DFSMS, etc.)
- S** Support programmer
- Z** Internal communication

Additional classes may be defined, but there is a maximum of 36 classes per command.

The legends for the prefix keywords and password requirement are located at the end of the table.

Table 2. DirMaint default command sets

Command Name	Prefix Keywords			PW Required	Command Set Class(es)	
	RTABP	FMAN	BSMPR		150A	140A
?	-----	---	---	-	...any...	...any...
ACCOUNT	RTABP	FMAN	BS-	y	..G.....	..G.....
ACIGROUP	RTABP	FMAN	BS-	y	A.....	.....
ACNTADD	RTABP	FMAN	BS-	y	..G.....	.....
ACNTDEL	RTABP	FMAN	BS-	y	..G.....	.....
ADD	RTABP	---N	**-	y	A.....	A.....
AMDISK	RTABP	FMAN	BS-	y	.D.....	.D.....
APPCPASS	RTABP	FMAN	BSP	y	..G.....	..G.....

## DirMaint Command Sets

Table 2. DirMaint default command sets (continued)

Command Name	Prefix Keywords			PW Required	Command Set Class(es)	
	RTABP	FMAN	BSMPR		150A	140A
ASSIGN <sup>1</sup>	RTABP	FM-	BS-	y	obsolete.	A.....S.
AUTH <sup>1</sup>	RTABP	FM-	BS-	y	obsolete.	.G.....
<sup>1</sup> The ASSIGN and AUTH commands have been replaced by the AUTHFOR command.						
AUTHBY	RTABP	FM-	BS-	y	.G.....	.....
AUTHFOR	RTABP	FM-	BS-	y	.G.....	.....
AUTHLINK	RTABP	FMA-	BS-	y	.G.....	.....
AUTHSCIF	RTABP	FM-	BS-	y	.G.....	.....
AUTOLOG	RTABP	FMAN	BS-	y	.G.....	A.....
BACKUP	RTABP	---	BSM	y	...H.O.S.	...H.O.S.
BATCH	RTABP	*-*N	-*M	y	.G.....	.G.....
CHECK	-----	---	-*M	-	...any...	.....
CHKSUM	RTABP	---	BS-	y	...H...S.	A..H...S.
CHNGID	RTABP	F--N	BS-	y	A.....	A.....
CHVADDR	RTABP	FMAN	BS-	y	.G.....	A.....
CLASS	RTABP	FMAN	BS-	y	A.....	.....
CLEANED	RTABP	FM-	BS-	y	obsolete.	.D.....Z
CLEANUP	RTABP	FMA-	BS-R	y	.D.....	.....
CLONEDISK	RTABP	FMAN	BS-	y	.D.....	.D.....
CMDISK	RTABP	FMAN	BS-	y	.D.....	.D.....
CMS	RTABP	---	BS-	y	.....S.	.....S.
COMMAND	RTABP	FMAN	BS-R	y	A.....	.....
CONSOLE <sup>2</sup>	RTABP	FMAN	BS-	y	.G.....	AD.H...S.
<sup>2</sup> In CMDLEVEL 140A, the CONSOLE command retrieves a copy of the DIRMAINT service machine's current console log. To obtain this function in CMDLEVEL 150A, use the GETCONSOLE command. In CMDLEVEL 150A, the CONSOLE command changes the virtual address or device type of a CONSOLE directory statement.						
COPIED	RTABP	FM-	BS-	y	obsolete.	.D.....Z
CP	RTABP	---	BS-	y	.....S.	.....S.
CPU	RTABP	FMAN	BS-	y	.G.....	.G.....
CRYPTO	RTABP	FMAN	BS-	y	A.....	.....
DASD	RTABP	---	BS-R	y	.D.H...S.	...H...S.
DASDOPT	RTABP	FMAN	BS-	y	AD.....	AD.....
DATAMOVE	RTABP	FMAN	BS-R	y	.....S.	AD.....S.
DATEFORMAT	RTABP	FMAN	BS-	y	.G.....	.....
DEDICATE	RTABP	FMAN	BS-	y	.....PS.	.....
DEFAULTS	-----	---	-*-	-	...any...	...any...
DEFINESTAG	RTABP	---	BS-	y	.....PS.	.....
DIRECT	RTABP	---	BS-	y	AD.H..PS.	AD.....S.
DIRECTORY	RTABP	---N	BS-	y	.....S.	.....
DIREEDIT	RTABP	---	BSM	y	AD.H..PS.	AD.H...S.

Table 2. DirMaint default command sets (continued)

Command Name	Prefix Keywords			PW Required	Command Set Class(es)	
	RTABP	FMAN	BSMPR		150A	140A
DIRMAP	RTABP	---	BSMR	y	AD.H..PS.	.....
DISABLE	RTABP	---	BS-	y	.....S.	.....S.
DISTRIB	RTABP	FMAN	BS-	y	..G.....	..G.....
DLINK	RTABP	FMAN	BS-	y	..G.....	..G.....
DMDISK	RTABP	FMAN	BS-	y	.D.....	.D.....
DMVCTL	-----	---	---	-	.....Z	.....
DROP <sup>3</sup>	RTABP	FM-	BS-	y	obsolete.	A.....S.
<sup>3</sup> The DROP and UNAUTH commands have been replaced by the DROPFOR command.						
DROPBY	RTABP	FM-	BS-	y	..G.....	.....
DROPFOR	RTABP	FM-	BS-	y	..G.....	.....
DROPSCIF	RTABP	FM-	BS-	y	..G.....	.....
DSATCTL	-----	---	---	-	.....Z	.....
DSECUSER	RTABP	FMAN	BS-	y	..G.....	.....
DUMP	RTABP	---	BS-R	y	...H...S.	...H...S.
D8ONECMD	RTABP	FMAN	BS-	y	A.....	.....
ELINK	RTABP	---	BS-	y	A.....	A.....
ENABLE	RTABP	---	BS-	y	.....PS.	.....S.
EXECDROP	-----	---	---	-	...any...	.....
EXECLOAD	-----	---	---	-	...any...	.....
EXTNCHK	RTABP	---	BS-	y	.....PS.	.D.....
FILE	RTABP	---	-S-	y	.....S.	.....S.
FREEXT	RTABP	---	BSMR	y	.D.H...S.	.D.H...S.
GET	RTABP	FM-	BSMR	y	A.....	A..H....
GETCONSOLE	RTABP	---	BSM	y	...H...S.	...H...S.
GLOBALOPTS	RTABP	--AN	BS-	y	.....P.	.....
GLOBALV	-----	---	-*-	-	...any...	...any...
HELP	-----	---	-*-	-	...any...	...any...
HISTORY	RTABP	---	BS-R	y	...H...S.	...H...S.
INCLUDE	RTABP	FMAN	BS-	y	A.....	.....
INITLZ	-----	---	BS-	-	.....Z	.....
INVEN	RTABP	---	BS-R	y	...H...S.	...H...S.
IOPRIORITY	RTABP	FMAN	BS-	y	A.....	.....
IPL	RTABP	FMAN	BS-	y	..G.....	..G.....
IUCV	RTABP	FMAN	BS-	y	A.....	.....
LINK	RTABP	FMAN	BSP	y	..G.....	..G.....
LOADDEV	RTABP	FMAN	BS-	y	..G.....	..G.....
LOCK	RTABP	FM-	BSM	y	A.....	A.....
LOGMSG	RTABP	---	BS-	y	AD.H...S.	AD.H...S.

Table 2. DirMaint default command sets (continued)

Command Name	Prefix Keywords			PW Required	Command Set Class(es)	
	RTABP	FMAN	BSMPR		150A	140A
LOGONBY	RTABP	FMAN	BS-	y	.G.....	.....
MACHINE	RTABP	FMAN	BS-	y	.G.....	.G.....
MAIL	RTABP	---	BS-	y	...H...S.	A.....S.
MAXSPOOL	RTABP	FMAN	BS-	y	A.....	A.....
MAXSTORAGE	RTABP	FMAN	BS-	y	A.....	A.....
MAXSTORE	RTABP	FMAN	BS-	y	A.....	A.....
MDAUDIT	RTABP	---	BS-R	y	AD..M....	AD..M....
MDISK	RTABP	FMAN	BSP	y	.G.....	.G.....
MDPW	RTABP	FMAN	BSM	y	.G.....	.G.....
MINIOPT	RTABP	FMAN	BS-	y	AD.....	AD.....
MMDISK	RTABP	--AN	BS-	y	.D...P..	.D...P..
NAMESAVE	RTABP	FMAN	BS-	y	A.....	A.....
NEEDPASS	R---P	---	-S-	y	.G.....	.G.....
NEWS	RTABP	---	BS-R	N	.G.....	.G.....
NICDEF	RTABP	FMAN	BS-	y	.G.....	.G.....
NOPDATA	RTABP	FMAN	BSM	y	.G.....	.G.....
NOTAPE	RTABP	---	BS-	y	....O.S.	....O.S.
OFFLINE	RTABP	---	BS-	y	AD....S.	AD....S.
ONLINE	RTABP	---	BSM	y	AD....S.	AD....S.
OPTION	RTABP	FMAN	BS-	y	.G.....	.G.....
PAVALIAS	RTABP	F-AN	BS-R	y	.G.....	.....
POOL	RTABP	FMAN	BS-	y	A.....	.....
POSIXFSROOT	RTABP	FMAN	BS-	y	.G.....	.....
POSIXGLIST	RTABP	FMAN	BS-	y	A.....	.....
POSIXGROUP	RTABP	--AN	BS-	y	A.....	.....
POSIXINFO	RTABP	FMAN	BS-	y	A.....	.....
POSIXIUPGM	RTABP	FMAN	BS-	y	.G.....	.....
POSIXIWDIR	RTABP	FMAN	BS-	y	.G.....	.....
POSIXOPT	RTABP	FMAN	BS-	y	A.....	.....
PRIORITY	RTABP	FMAN	BS-	y	A.....	.....
PRIOSET	RTABP	FMAN	BS-	y	A.....	A.....
PRIVCLAS	RTABP	FMAN	BS-	y	.....	A.....
PRIVCLASS	RTABP	FMAN	BS-	y	A.....	.....
PURGE	RTABP	FM-N	BSM	y	A.....	A.....
PW	RTABP	FM-	BSMP	y	.G.....	.G.....
PW? <sup>4</sup>	RTABP	FM-	BSM	?	.G.....	.G.....

<sup>4</sup> When no administratively controlled password change is pending, you are not normally prompted to enter a password for the PW? command. However, if prompted, your entry will not be verified. When an administratively controlled password change is pending, you will be prompted to enter a password that will be verified before the system responds with your new password.

Table 2. DirMaint default command sets (continued)

Command Name	Prefix Keywords			PW Required	Command Set Class(es)	
	RTABP	FMAN	BSMPR		150A	140A
PWGEN	RTABP	---	BS-	y	....M....	....M...S.
PWMON	RTABP	---	BS-R	y	....M....	....M...S.
QLOG	RTABP	---	BS-R	y	..G.....	..G.....
QRY	RTABP	---	BS-	y	AD.H...S.	AD.H...S.
QUERY	RTABP	---	BS-	y	AD.H...S.	AD.H...S.
REPLACE	RTABP	F--N	BS-	y	.....S.	A.....S.
REVIEW	RTABP	FM-	BSMR	y	..G.....	..G.....
RLDCODE	RTABP	---	BS-	y	.....S.	.....S.
RLDDATA	RTABP	---	BS-	y	.....S.	.....S.
RLDXTN	RTABP	---	BS-	y	.....S.	.....S.
RMDISK	RTABP	FMAN	BS-	y	.D...P..	.D...P..
SATELLITE	RTABP	FMAN	BS-	y	.....S.	.....
SCAN	RTABP	---	BS-R	y	...H...S.	A..H....
SCREEN	RTABP	FMAN	BS-	y	..G.....	..G.....
SECUSER	RTABP	FMAN	BS-	y	..G.....	.....
SEND	RTABP	---	BS-R	y	...H...S.	.....S.
SETACNT	RTABP	FMAN	BS-	y	A.....	.....
SETCLASS	RTABP	FMAN	BS-	y	A.....	.....
SETCPU	RTABP	FMAN	BS-	y	A.....	A.....
SETMACH	RTABP	FMAN	BS-	y	A.....	A.....
SETOPTN <sup>5</sup>	RTABP	FMAN	BS-	y	A.....	..G.....
<sup>5</sup> In CMDLEVEL 140A, the SETOPTN command manipulates the user's DirMaint control options. To obtain this function in CMDLEVEL 150A, use the USEROPTN command. In CMDLEVEL 150A, the SETOPTN command allows an administrator to add or change values on a user's OPTION directory statement.						
SETPRIORITY	RTABP	FMAN	BS-	y	A.....	.....
SETPW	RTABP	FMAN	BS-	y	A...M....	A...M....
SETSTAG	RTABP	FMAN	BS-	y	A.....	.....
SHARE	RTABP	FMAN	BS-	y	A.....	.....
SHUTDOWN	RTABP	---	BS-	y	....O.S.	....O.S.
SPECIAL	RTABP	FMAN	BS-	y	..G.....	.....
SPOOL	RTABP	FMAN	BS-	y	..G.....	.....
SSI	RTABP	---N	BS-R	y	A.....S.	.....
STAG	RTABP	FMAN	BS-	y	..G.....	.....
STATUS	RTABP	---	BS-	y	...H...S.	...H...S.
STDEVOPT	RTABP	FMAN	BS-	y	A.....	.....
STORAGE	RTABP	FMAN	BS-	y	..G.....	..G.....
SUBSCRIBE	RTABP	FM-	BS-	y	A...P..	.....
SUPGLIST	RTABP	FMAN	BS-	y	..G.....	.....
SYSAFFIN	RTABP	FM-	BS-	y	A.....	A.....

## DirMaint Command Sets

Table 2. DirMaint default command sets (continued)

Command Name	Prefix Keywords			PW Required	Command Set Class(es)	
	RTABP	FMAN	BSMPR		150A	140A
TERM	RTABP	FM-	BS-	y	..G.....	..G.....
TESTPW	RTABP	FM-	BSMP	y	..G.....	..G.....
TMDISK	RTABP	FMAN	BS-	y	A.....	A.....
UNAUTH <sup>6</sup>	RTABP	FM-	BS-	y	obsolete.	..G.....
<sup>6</sup> The UNAUTH and DROP commands have been replaced by the DROPFOR command.						
UNDOSSI	RTABP	---N	BS-R	y	A.....S.	.....
UNLOCK	RTABP	FM-	BS-	y	A.....	A.....
USEDEXT	RTABP	---	BSMR	y	.D.H...S.	...H...S.
USER	RTABP	---	BS-	y	...H...S.	...H...S.
USERMAP	RTABP	---	BS-R	y	...H...S.	...H...S.
USEROPTN	RTABP	FM-	BS-	y	..G.....	..G.....
VERIFIED	RTABP	FM-	BS-	y	obsolete.	..G.....
VMRELOCATE	RTABP	FMAN	BS-	y	A.....	.....
WORKUNIT	RTABP	---	BS-	y	.D.....	AD.....
XAUTOLOG	RTABP	FMAN	BS-	y	..G.....	.....
XCONFIG	RTABP	FMAN	BS-	y	A.....	.....
ZAPMDISK	RTABP	FMAN	BS-	y	.....Z	.....

Prefix Keyword Legend. Keywords not honored are ignored.

The RTABP group:

**R**

The REQuest keyword is honored.

**T**

The TOnode or TOsys keyword is honored.

**A**

The ASuser keyword is honored.

**B**

The BYuser keyword is honored.

**P**

The PRESET keyword is honored in command level 150A.

The FMAN group:

**F**

The FORuser keyword is honored.

**M**

The MULTIuser keyword is honored in command level 150A.

**A**

The ATnode or ATsys keyword is honored.

**\***

The FORuser and ATnode or ATsys keywords may be used, subject to restrictions or usage notes.

**N**

The NODIRectxa keyword is honored.

The BSMPR group:

**B**

The command may be used within a BATCH file.

**\***

The command can be used within a BATCH file, subject to restrictions or usage notes.

**S**

The command may be used through DVHSAPI.

**\***

The command can be used through DVHSAPI, subject to restrictions or usage notes.

**M**

The command has optional parameters available, but none of them are required; use the MENU keyword to force display of the menu. The PROMPT keyword has no effect on this command.

**P**

The command contains optional passwords for which defaults will be used if not specified; use PROMPT to be prompted for omitted passwords. The MENU keyword has no effect on this command.

**-**

Neither the MENU nor the PROMPT keywords have any effect on this command. The command either has no parameters, or at least one parameter is required, with none of the parameters being a password.

**R**

The RETurnvia/RETVia keyword is honored.

Password Required Legend.

**Y**

Password verification is the IBM supplied default; although this may be tailored by the installation or waived by the user.

**N**

Password verification is NOT required.

**?**

Password verification is USUALLY not required, unless an administrator assigned password change is pending.

**-**

The command is processed in the user's virtual machine; password verification is not applicable.



## Appendix B. Device Characteristics

Table 3. Maximum Sizes and Blocks Per Cylinder for Supported CKD Devices

Disk Type	Maximum CKD Minidisk Sizes		Blocks per CKD Cylinder with a block size of:				
	800- Byte CMS Format	Other Usage	800	512	1K	2K	4K
3380-01	121	885	540	690	465	270	150
3380-02	121	1770	540	690	465	270	150
3380-03	121	2655	540	690	465	270	150
3380-91	N/S	1113	540	690	465	270	150
3380-92	N/S	2226	540	690	465	270	150
3380-93	N/S	3339	540	690	465	270	150
3380-99	N/S	10017	540	690	465	270	150
3380-459	N/S	1459	540	690	465	270	150
3390-01	N/S	1113	N/S	735	495	315	180
3390-02	N/S	2226	N/S	735	495	315	180
3390-03	N/S	3339	N/S	735	495	315	180
3390-09	N/S	10017	N/S	735	495	315	180
3390-084	N/S	1084	N/S	735	495	315	180
3390-151	N/S	2226	N/S	735	495	315	180
3390-153	N/S	4365	N/S	735	495	315	180
3390-455	N/S	455	N/S	735	495	315	180
3390-568	N/S	1568	N/S	735	495	315	180
3390-32K	N/S	32760	N/S	735	495	315	180
3390-64K	N/S	65520	N/S	735	495	315	180
3390-A	N/S	268434453	N/S	735	495	315	180

**Note:**

1. N/S = NOT supported.
2. For a 3390 used in 3380 compatibility mode, specify:

Table 4. Specifications for a 3390 used in 3380 compatibility mode

Disk Type	Cylinders
<b>3380-91</b>	1113
<b>3380-92</b>	2226
<b>3380-93</b>	3339
<b>3380-99</b>	10017

3. For a 9392, specify 3390.

## Device Characteristics

4. For a 9395, specify either 3390-3 (native) or 3380-3 (emulation).
5. For a 3995-151, specify 3390-151. For a 3995-153, specify 3390-153.
6. For a Multiprise® 2000 internal disk (or any other DASD) emulating a 3380, specify:

Table 5. Specifications for a Multiprise 2000 internal disk (or any other DASD) emulating a 3380

Disk Type	Cylinders
<b>3380-01</b>	885
<b>3380-02</b>	1770
<b>3380-03</b>	2655
<b>3380-459</b>	1459

7. For a Multiprise 2000 internal disk (or any other DASD) emulating a 3390, specify:

Table 6. Specifications for a Multiprise 2000 internal disk (or any other DASD) emulating a 3390

Disk Type	Cylinders
<b>3390-01</b>	1113
<b>3390-02</b>	2226
<b>3390-03</b>	3339
<b>3390-09</b>	10017
<b>3390-32K</b>	32760
<b>3390-64K</b>	65520
<b>3390-455</b>	455
<b>3390-568</b>	1568

8. For any other DASD that emulates a VM-supported disk device with a non-standard capacity, assign your own *model number*. Simply update the AUTOBLOCK and DEFAULTS sections of the EXTENT CONTROL file to include your new *model(s)* of the emulated device

Table 7. Maximum Sizes and FB-512 Blocks Per CMS Block for Supported FBA Devices

Disk Type	Maximum FBA Minidisk Sizes		FB-512 Blocks Per Logical CMS Block with a size of:				
	800-Byte CMS Format	Other Usage	800	512	1K	2K	4K
9336-020	N/S	1672881	N/S	1	2	4	8
9336-381	N/S	799999992	N/S	1	2	4	8
9336-1T	N/S	2147483640	N/S	1	2	4	8
V-DISK	N/S	4194296	N/S	1	2	4	8

### Note:

1. N/S = NOT supported.
2. The above device types may be specified as either the actual device type or as FB-512.
3. For a virtual disk in storage (V-DISK), the device type must be specified as FB-512.
4. CP allocates virtual disks in storage and FBA disks in 8-block pages. Therefore, the size of the V-DISK or FBA disk may be rounded up to the nearest page.

5. z/VM supports a SCSI disk, which is emulated as a 9336-020, up to a capacity of 2147483640 512-byte blocks (1 TB). Note, however, that directory, paging, and spooling allocations must reside within the first 16777216 blocks (blocks 0 to 16777215) of a CP-formatted disk.

Table 8. Maximum Sizes and Blocks Per Cylinder for Obsolete CKD Devices

Disk Type	Maximum CKD Minidisk Sizes		Blocks per CKD Cylinder with a block size of:				
	800-Byte CMS Format	Other Usage	800	512	1K	2K	4K
2305-1	N/S	48	N/S	N/S	N/S	N/S	N/S
2305-2	N/S	96	N/S	N/S	N/S	N/S	N/S
231x	203	203	150	220	120	60	30
3330-01	246	404	266	380	209	114	57
3330-11	246	808	266	380	209	114	57
3340-35	348	348	96	144	84	42	24
3340-70	682	696	96	144	84	42	24
3350	115	555	570	810	450	240	120
3375	182	959	360	480	300	168	96
9345-01	N/S	1440	N/S	615	420	255	150
9345-02	N/S	2156	N/S	615	420	255	150

**Note:**

1. N/S = NOT supported.
2. All of these devices are obsolete and are no longer supported. Capacity information is shown above for migration planning only.
3. For a 3333, see 3330. For a 3344, see 3340.
4. For a 3350 used in 3330 compatibility mode, see 3330-1.

Table 9. Maximum Sizes and FB-512 Blocks Per CMS Block for Obsolete FBA Devices

Disk Type	Maximum FBA Minidisk Sizes		FB-512 Blocks Per Logical CMS Block with a size of:				
	800-Byte CMS Format	Other Usage	800	512	1K	2K	4K
3310	N/S	12616	N/S	1	2	4	8
3370-01	N/S	558000	N/S	1	2	4	8
3370-02	N/S	712752	N/S	1	2	4	8
9313	N/S	246420	N/S	1	2	4	8
9332-400	N/S	360036	N/S	1	2	4	8
9332-600	N/S	554800	N/S	1	2	4	8
9335	N/S	804714	N/S	1	2	4	8
9336-010	N/S	920115	N/S	1	2	4	8
0671-00	N/S	574560	N/S	1	2	4	8
0671-04	N/S	624456	N/S	1	2	4	8

## Device Characteristics

Table 9. Maximum Sizes and FB-512 Blocks Per CMS Block for Obsolete FBA Devices (continued)

Disk Type	Maximum FBA Minidisk Sizes		FB-512 Blocks Per Logical CMS Block with a size of:				
	800-Byte CMS Format	Other Usage	800	512	1K	2K	4K
0671-08	N/S	513072	N/S	1	2	4	8

**Note:**

1. N/S = NOT supported.
2. All of these devices are obsolete and are no longer supported. Capacity information is shown above for migration planning only.
3. The 0671 DASD was also known as a 9371.

## Appendix C. The Synchronous Application Programming Interface (SAPI)

The Synchronous Application Programming Interface (SAPI) provides a method for exec programs to parse message responses from DirMaint commands. When this interface is activated, key information from the message response is returned in a structured and predictable format that allows a program to parse the fields of the response, analyze the information, and take any needed action.

### Sample Programs

The DIRMSAPI EXEC is an example of a REXX program that sets up the proper environment for using the SAPI interface. The DVHSAPI EXEC issues a command to the DIRMAINT service machine and waits for the response, which it passes back to DIRMSAPI EXEC. Both of these sample exec files reside on the user interface disk (4VMDVH10 11F, by default) as file names DIRMSAPI EXEC SAMP and DVHSAPI EXEC SAMP. These files are available for your examination.

Given a DirMaint command string, the DIRMSAPI EXEC routine calls the DVHSAPI EXEC to send the command to the DIRMAINT service machine and wait for the responses from the service machine. The responses are returned from DVHSAPI in the "DVHSAPI." stem variable.

The DVHSAPI EXEC routine is not intended to be invoked as a command directly from the console. It is intended to be called by a customer supplied REXX program. The DIRMSAPI EXEC (or DIRMSAPI EXEC SAMP file) is a sample of such a program.

### Applied SAPI Coding Rules

The following rules have been applied to the DIRMSAPI EXEC sample.

- An EXEC DIRMAINT EXECLOAD command should be issued before calling DVHSAPI, and an EXEC DIRMAINT EXECDROP command should be issued when no further calls to DVHSAPI are anticipated for a significant time interval. When issuing multiple commands, a single EXEC DIRMAINT EXECLOAD command should be used before the group of calls to DVHSAPI, and one EXEC DIRMAINT EXECDROP command should be used after the group. Service machines that frequently issue DIRMAINT commands should include an EXEC DIRMAINT EXECLOAD command in their PROFILE EXEC.
- If the PRESET global variable, in the DVH15 pool, has a value other than blanks or NOLOG, DirMaint will send that value to the DIRMAINT server without prompting for a password. To prevent disclosure of that password, the variable will be reset to a value of NOLOG upon exit, thus the value must be set prior to each call to the DIRMAINT EXEC.

**Note:** If an EXEC DIRMAINT EXECLOAD command has already been issued and the files are still loaded, it is prudent to issue an EXEC DIRMAINT EXECDROP command before re-issuing the EXEC DIRMAINT EXECLOAD command. It does no harm to always issue an EXEC DIRMAINT EXECDROP command before issuing any EXEC DIRMAINT EXECLOAD command.

```
/* Miniature sample for use of DVHSAPI. */
Address 'COMMAND'
Trace 'C'
'EXEC DIRMAINT EXECLOAD'
'EXEC DVHSAPI FOR USER1 DISTRIB NEWDIST1'
'PIPE STEM DVHSAPI. | CONSOLE'
'EXEC DVHSAPI FOR USER2 DISTRIB NEWDIST2'
'PIPE STEM DVHSAPI. | CONSOLE'
'EXEC DVHSAPI FOR USER3 DISTRIB NEWDIST3'
'PIPE STEM DVHSAPI. | CONSOLE'
'EXEC DVHSAPI FOR USER4 DISTRIB NEWDIST4'
'PIPE STEM DVHSAPI. | CONSOLE'
```

```
'EXEC DIRMAINT EXECDROP'
Exit
```

- Calls to DVHSAPI may optionally preface the request with a time-out limit between 10 seconds (+00:00:1) and just under 12 hours (+11:59:5). The default is 15 seconds (+00:15), if not specified.
- The return code from DVHSAPI indicates the status of your request.

**1191**

Message DVH1191I was expected but not received. Your command has not been sent to the DIRMAINT service machine. The messages passed back in the DVHSAPI stem variable explain why.

**1196**

The DVHSAPI EXEC exceeded the *interval* second time limit while expecting message DVHnnnn. The responses in the DVHSAPI stem variable are incomplete.

**2288**

Message DVH2288I was expected but not received. The DIRMAINT service machine received your request, but has rejected it from further processing. The messages passed back in the DVHSAPI stem variable explain why.

**2289**

Message DVH2289E was received. The DIRMAINT service machine processed your request, which ended with a non-zero return code. The messages passed back in the DVHSAPI stem variable explain why.

**0**

Message DVH2289I was received. The DIRMAINT service machine processed your request, which ended with a zero return code. The DVHSAPI stem variable will contain the messages, if any, generated while processing your request.

**xxxx**

Other non-zero return codes indicate one of three things:

1. The DIRMAINT service machine failed to respond to your request within the allotted time interval; or
  2. The DVHSAPI EXEC was confused by messages from the DIRMAINT service machine generated by a previous request, or by spontaneous messages from the DIRMAINT service machine; or
  3. Some other error condition was encountered that the DVHSAPI EXEC could not handle.
- Be aware that some commands result in asynchronous processing. For example, an EXEC DVHSAPI +00:05:0 BATCH SETALL MYPWS command will send the contents of the SETALL MYPWS file to the DIRMAINT service machine for processing, with message DVH2289I and a zero return code indicating that the file has been received and appended to the batch queue for processing. A zero return code from DVHSAPI in this case does not indicate that any or all of the requests within the file have been processed, or that they will all be processed successfully. For example, if one of the records in the file is FOR DVHTEST PW DVHTEST, this request will very likely fail with message DVH2924E, even though you have already received message DVH2289I and a zero return code from your BATCH command.
  - The DVHSAPI\_END\_MSG. CONFIG\* DATADVH entry can be used to identify when the DVHSAPI routine exits and returns control back to the calling application. The default is to end when message DVHREQ2289I is received and no DVHSAPI\_END\_MSG. entries are specified. The sample CONFIG SAMPDVH file contains these entries:

```
- DVHSAPI_END_MSG.DVHSCU3541I= DVHREQ2289I DVHSHN3430I
```

```
- DVHSAPI_END_MSG.DVHPUR3370I= DVHREQ2289I DVHXXX3525I
```

The message DVHSAPI\_END\_MSG.DVHSCU3541I= DVHREQ2289I DVHSHN3430I tells DVHSAPI that if it receives the message DVHSCU3541I, it should wait for both message DVHREQ2289I and DVHSHN3430I to be received, in any order, before determining whether the command finished successfully.

You can use "XXX" for the issuing module ID portion of any ending message identifier (i.e. after the equal sign) to indicate that the message may be issued from any module/EXEC. For example, the statement:

DVHSAPI\_END\_MSG.DVHPUR3370I= DVHREQ2289I DVHXXX3525I

tells DVHSAPI that if it receives the message DVHPUR3370I, it should wait for both message DVHREQ2289I and for any message that begins with "DVH" and ends with "3525I" before determining whether the command finished successfully.

**Note:** This configuration entry can be used for any other condition requiring multiple messages to be received when determining if the command completed successfully.

- The DVHSAPI EXEC does not support use of the TOsys or TOnode prefix operand for remote networking. When used within a multi-system CSE cluster, cross-system SMSG capability must be enabled.



## 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.

## Programming Interface Information

---

This book documents intended Programming Interfaces that allow the customer to write programs to obtain services of DirMaint.

## 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](http://www.ibm.com/legal/copytrade) (<https://www.ibm.com/legal/copytrade>).

The registered trademark Linux<sup>®</sup> is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide 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-Express Customer's Guide and Reference](#) (<https://www.ibm.com/support/pages/node/6019492>), SA22-7935
- [Open Systems Adapter-Express Integrated Console Controller User's Guide](#) (<https://www.ibm.com/support/pages/node/6019810>), SC27-9003
- [Open Systems Adapter-Express Integrated Console Controller 3215 Support](#) ([https://www.ibm.com/docs/en/SSLTBW\\_2.1.0/com.ibm.zos.v2r1.ioa/ioa.htm](https://www.ibm.com/docs/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.ioa/ioa.htm)), SA23-2247
- [Open Systems Adapter/Support Facility on the Hardware Management Console](#) ([https://www.ibm.com/docs/en/SSLTBW\\_2.1.0/com.ibm.zos.v2r1.ioa/ioa.htm](https://www.ibm.com/docs/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.ioa/ioa.htm)), SC14-7580

## Performance Toolkit for z/VM

- [z/VM: Performance Toolkit Guide](#), SC24-6302
- [z/VM: Performance Toolkit Reference](#), SC24-6303

## 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/servers/resourcelink/svc00100.nsf/pages/zosv2r5gc350033/\\$file/ickug00\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5gc350033/$file/ickug00_v2r5.pdf)), GC35-0033

### Environmental Record Editing and Printing Program

- Environmental Record Editing and Printing Program (EREP): Reference ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5gc350152/\\$file/ifc2000\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5gc350152/$file/ifc2000_v2r5.pdf)), GC35-0152
- Environmental Record Editing and Printing Program (EREP): User's Guide ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5gc350151/\\$file/ifc1000\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5gc350151/$file/ifc1000_v2r5.pdf)), GC35-0151

## Related Products

---

### z/OS

- *Common Programming Interface Communications Reference* (<https://publibfp.dhe.ibm.com/epubs/pdf/c2643999.pdf>), SC26-4399
- z/OS and z/VM: Hardware Configuration Definition Messages ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sc342668/\\$file/cbdrm100\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sc342668/$file/cbdrm100_v2r5.pdf)), SC34-2668
- z/OS and z/VM: Hardware Configuration Manager User's Guide ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sc342670/\\$file/eequ100\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sc342670/$file/eequ100_v2r5.pdf)), SC34-2670
- z/OS: Network Job Entry (NJE) Formats and Protocols ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa320988/\\$file/hasa600\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa320988/$file/hasa600_v2r5.pdf)), SA32-0988
- z/OS: IBM Tivoli Directory Server Plug-in Reference for z/OS ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa760169/\\$file/glpa300\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa760169/$file/glpa300_v2r5.pdf)), SA76-0169
- z/OS: Language Environment Concepts Guide ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380687/\\$file/ceea800\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380687/$file/ceea800_v2r5.pdf)), SA38-0687
- z/OS: Language Environment Debugging Guide ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5ga320908/\\$file/ceea100\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5ga320908/$file/ceea100_v2r5.pdf)), GA32-0908
- z/OS: Language Environment Programming Guide ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380682/\\$file/ceea200\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380682/$file/ceea200_v2r5.pdf)), SA38-0682
- z/OS: Language Environment Programming Reference ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380683/\\$file/ceea300\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380683/$file/ceea300_v2r5.pdf)), SA38-0683
- z/OS: Language Environment Runtime Messages ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380686/\\$file/ceea900\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380686/$file/ceea900_v2r5.pdf)), SA38-0686
- z/OS: Language Environment Writing Interlanguage Communication Applications ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380684/\\$file/ceea400\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa380684/$file/ceea400_v2r5.pdf)), SA38-0684
- z/OS: MVS Program Management Advanced Facilities ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa231392/\\$file/ieab200\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa231392/$file/ieab200_v2r5.pdf)), SA23-1392
- z/OS: MVS Program Management User's Guide and Reference ([https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa231393/\\$file/ieab100\\_v2r5.pdf](https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r5sa231393/$file/ieab100_v2r5.pdf)), SA23-1393

## **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](#)



---

# Index

## A

ACCOUNT command [14](#)  
ACIGROUP command [16](#)  
ACNTADD command [17](#)  
ACNTDEL command [19](#)  
ADD command [20](#)  
AMDISK command [25](#)  
APPCPASS command [31](#)  
AUTHBY command [33](#)  
AUTHFOR command [34](#)  
AUTHLINK command [35](#)  
AUTHSCIF command [37](#)  
AUTOLOG command [38](#)

## B

BACKUP command [40](#)  
BATCH command [42](#)

## C

CHECK command [44](#)  
CHKSUM command [46](#)  
CHNGID command [47](#)  
CHVADDR command [49](#)  
CLASS command [50](#)  
CLEANUP command [51](#)  
CLONEDISK command [54](#)  
CMDISK command [59](#)  
CMS command [65](#)  
COMMAND command [66](#)  
CONSOLE command [67](#)  
CP command [69](#)  
CPU command [70](#)  
CRYPTO command [72](#)

## D

D8ONECMD command [108](#)  
DASD command [74](#)  
DASD devices [307](#)  
DASDOPT command [79](#)  
DATAMOVE command [81](#)  
DATEFORM command [83](#)  
DEDICATE command [84](#)  
DEFAULTS command [86](#)  
DEFINESTAG command [90](#)  
DIRECT command [92](#)  
DIRECTORY command [93](#)  
DIREDIR command [95](#)  
DIRM exec [6](#)  
DIRMAINT exec [6](#)  
DIRMAP command [96](#)  
DISABLE command [98](#)  
DISTRIB command [99](#)

DLINK command [100](#)  
DMDISK command [101](#)  
DROPBY command [103](#)  
DROPFOR command [104](#)  
DROPSCIF command [105](#)  
DSECUSER command [106](#)  
DUMP command [107](#)

## E

ELINK command [109](#)  
ENABLE command [111](#)  
EXECDROP command [112](#)  
EXECLOAD command [113](#)  
EXTNCHK command [114](#)

## F

FILE command [115](#)  
FREEEXT command [117](#)

## G

GET command [120](#)  
GETCONSOLE command [121](#)  
GLOBALOPTS command [122](#)  
GLOBALV command [124](#)

## H

HELP command [128](#)  
HISTORY command [130](#)

## I

INCLUDE command [131](#)  
INVEN command [132](#)  
IOPRIORITY command [133](#)  
IPL command [134](#)  
IUCV command [136](#)  
IVP command [139](#)

## L

LINK command [144](#)  
LOADDEV command [147](#)  
LOCK command [149](#)  
LOGMSG command [150](#)  
LOGONBY command [151](#)

## M

MACHINE command [152](#)  
MAIL command [154](#)  
MAXSPOOL command [155](#)

MAXSTORAGE command [156](#)  
MAXSTORE command [156](#)  
MDAUDIT command [158](#)  
MDISK command [162](#)  
MDPW command [165](#)  
message examples, notation used in [4](#)  
MINIOPT command [166](#)  
MMDISK command [168](#)

## N

NAMESAVE command [170](#)  
NEEDPASS command [171](#)  
NEWS command [172](#)  
NICDEF command [173](#)  
NOPDATA command [177](#)  
NOTAPE command [178](#)  
notation used in message and response examples [4](#)

## O

OFFLINE command [179](#)  
ONLINE command [180](#)  
OPTION command [181](#)

## P

PAVALIAS command [186](#)  
POOL command [188](#)  
POSIXFSROOT command [189](#)  
POSIXGLIST command [190](#)  
POSIXGROUP command [192](#)  
POSIXINFO command [193](#)  
POSIXIWDIR command [196](#)  
POSIXOPT command [197](#)  
POSIXUPGM command [195](#)  
PRIORITY command [199](#)  
PRIOSET command [200](#)  
PRIVCLASS command [201](#)  
PURGE command [202](#)  
PW command [204](#)  
PW? command [206](#)  
PWGEN command [207](#)  
PWMON command [209](#)

## Q

QLOG command [211](#)  
QRY command [212](#)  
QUERY command [212](#)

## R

REPLACE command [213](#)  
response examples, notation used in [4](#)  
REVIEW command [215](#)  
RLDCODE command [216](#)  
RLDDATA command [217](#)  
RLDEXTN command [218](#)  
RMDISK command [220](#)

## S

SATELLITE command [224](#)  
SCAN command [226](#)  
SCREEN command [230](#)  
SECUSER command [232](#)  
SEND command [234](#)  
SETACNT command [235](#)  
SETCLASS command [237](#)  
SETCPU command [238](#)  
SETMACH command [240](#)  
SETOPTN command [242](#)  
SETPRIORITY command [247](#)  
SETPW command [248](#)  
SETSTAG command [250](#)  
SHARE command [252](#)  
SHUTDOWN command [254](#)  
SPECIAL command [255](#)  
SPOOL command [258](#)  
SSI command [260](#)  
STAG command [261](#)  
STATUS command [263](#)  
STDEVOPT command [265](#)  
STORAGE command [266](#)  
SUBSCRIBE command [268](#)  
SUPGLIST command [270](#)  
Synchronous Application Programming Interface (SAPI)  
DIRMSAPI EXECSTAMP [311](#)  
rules [311](#)  
sample programs [311](#)  
syntax diagrams, how to read [2](#)  
SYSAFFIN command [272](#)

## T

TERM command [273](#)  
TESTPW command [275](#)  
TMDISK command [277](#)

## U

UNDOSSI command [279](#)  
UNLOCK command [280](#)  
USEDEXT command [281](#)  
USER command [284](#)  
USERMAP command [286](#)  
USEROPTN command [287](#)

## V

VMRELOCATE command [291](#)

## W

WORKUNIT command [293](#)

## X

XAUTOLOG command [295](#)  
XCONFIG command [297](#)





Product Number: 5741-A09

Printed in USA

SC24-6281-73

