



**Program Directory for  
IBM TCP/IP for VM  
and Features**

Version 2  
Release 4

Program Number 5735-FAL

for Use with  
VM/ESA Version 1 Release 2.2  
VM/ESA Version 2 Release 1.0  
VM/ESA Version 2 Release 2.0

Document Date: December, 1999

xxxx-yyyy-zz

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xiv.

This program directory, dated December, 1999, applies to TCP/IP Version 2 Release 4 for VM (TCP/IP V2 R4 for VM), Program Number 5735-FAL.

A form for reader's comments appears at the back of this publication. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1990, 1998. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Notices</b> . . . . .	xiv
Trademarks and Service Marks . . . . .	xiv
<b>1.0 Introduction</b> . . . . .	1
1.1 Program Description . . . . .	2
<b>2.0 Program Materials</b> . . . . .	4
2.1 Basic Machine-Readable Material . . . . .	4
2.2 Optional Machine-Readable Material . . . . .	5
2.2.1 Optional Machine-Readable Material for Source Feature . . . . .	5
2.2.2 Optional Machine-Readable Material for NFS Feature . . . . .	6
2.2.3 Optional Machine-Readable Material for Kerberos US Feature . . . . .	7
2.2.4 Optional Machine-Readable Material for Kerberos WT Feature . . . . .	8
2.3 Program Publications . . . . .	9
2.3.1 Basic Program Publications . . . . .	9
2.3.2 Optional Program Publications . . . . .	9
2.3.3 Softcopy Publications . . . . .	9
2.4 Microfiche Support . . . . .	10
2.5 Publications Useful During Installation and Service . . . . .	10
<b>3.0 Program Support</b> . . . . .	11
3.1 Preventive Service Planning . . . . .	11
3.2 Statement of Support Procedures . . . . .	11
3.3 Service Information . . . . .	12
3.3.1 Problem Documentation . . . . .	12
3.3.2 Communicating Your Comments to IBM . . . . .	13
<b>4.0 Program and Service Level Information</b> . . . . .	14
4.1 Program Level Information - TCP/IP V2 R4 for VM . . . . .	14
4.2 Program Level Information - NFS Feature . . . . .	15
4.3 Program Level Information - Kerberos US Feature . . . . .	15
4.4 Program Level Information - Kerberos WT Feature . . . . .	15
4.5 Service Level Information . . . . .	15
4.6 Cumulative Service (RSU) Tape Information . . . . .	15
<b>5.0 Installation Requirements and Considerations</b> . . . . .	16
5.1 Hardware Requirements . . . . .	16
5.2 Program Considerations . . . . .	16
5.2.1 Operating System Requirements . . . . .	16
5.2.2 Other Program Product Requirements . . . . .	17
5.2.2.1 Other Program Product Requirements - TCP/IP V2 R4 for VM . . . . .	17
5.2.2.2 Other Program Product Requirements - Features . . . . .	18

5.2.3	Program Installation/Service Considerations	18
5.3	DASD Storage and User ID Requirements	20
5.3.1	DASD Requirements for TCP/IP V2 R4 for VM	23
5.3.2	DASD Requirements for Source Feature	27
5.3.3	DASD Storage Requirements for IBM Network Station Support	28
5.3.3.1	DASD Storage Requirements for Target Minidisks	28
5.3.3.2	DASD Storage Requirements for Existing Minidisks	29
5.3.4	DASD Storage Requirements for DHCPD Server Support	29
5.3.4.1	DASD Storage Requirements for Target Minidisks	29
5.3.4.2	DASD Storage Requirements for Existing Minidisks	31
5.3.5	DASD Requirements for NFS Feature	32
5.3.6	DASD Requirements for Kerberos US Feature	34
5.3.7	DASD Requirements for Kerberos WT Feature	36
<b>6.0</b>	<b>Installation Instructions</b>	<b>38</b>
6.1	TCP/IP for VM Installation Process Overview	38
6.2	Installing TCP/IP V2 R4 for VM	40
6.2.1	Plan Your Installation For TCP/IP V2 R4 for VM	40
6.2.2	Allocate Resources for Installing TCP/IP V2 R4 for VM	43
6.2.2.1	PROFILE Directory Entries	44
6.2.2.2	Installing TCP/IP V2 R4 for VM to Minidisks	44
6.2.2.3	Installing TCP/IP V2 R4 for VM to SFS Directories	46
6.2.3	Install TCP/IP V2 R4 for VM	49
6.2.3.1	Update Build Status Table for TCP/IP V2 R4 for VM	55
6.2.4	Install the Initial TCP/IP V2 R4 for VM RSU	55
6.2.5	Place TCP/IP V2 R4 for VM Into Production	55
6.2.5.1	Copy TCP/IP V2 R4 for VM Files Into Production	55
6.2.5.2	Configure TCP/IP V2 R4 for VM	57
6.2.5.3	TCP/IP V2 R4 for VM Product and Sample Configuration Files	58
6.3	Installing TCP/IP V2 R4 for VM - Source Feature	62
6.3.1	Plan Your Installation for the Source Feature	62
6.3.2	Allocate Resources for Installing the Source Feature	65
6.3.2.1	Installing the Source Feature to Minidisks	65
6.3.2.2	Installing the Source Feature to SFS Directories	65
6.3.3	Install the Source Feature	66
6.3.4	Place the Source Feature Into Production	68
6.3.4.1	Unpack the Source Feature	68
6.4	Installing TCP/IP V2 R4 for VM - NFS Feature	69
6.4.1	Plan Your Installation for the NFS Feature	69
6.4.2	Allocate Resources for Installing the NFS Feature	72
6.4.2.1	PROFILE Directory Entries	73
6.4.2.2	Installing the NFS Feature to Minidisks	73
6.4.2.3	Installing the NFS Feature to SFS Directories	74
6.4.3	Install the NFS Feature	75
6.4.3.1	Update Build Status Table for the NFS Feature	78
6.4.4	Place the NFS Feature Into Production	78

6.4.4.1	Copy NFS Feature Files Into Production	78
6.4.4.2	Configure the NFS Feature	80
6.4.4.3	NFS Feature Runtime and Sample Configuration Files	82
6.4.4.4	Unpack Source Files	83
6.5	Installing TCP/IP V2 R4 for VM - Kerberos US Feature	84
6.5.1	Plan Your Installation for the Kerberos US Feature	84
6.5.2	Allocate Resources for Installing the Kerberos US Feature	87
6.5.2.1	PROFILE Directory Entries	88
6.5.2.2	Installing the Kerberos US Feature to Minidisks	88
6.5.2.3	Installing the Kerberos US Feature to SFS Directories	89
6.5.3	Install the Kerberos US Feature	91
6.5.3.1	Update Build Status Table for the Kerberos US Feature	94
6.5.4	Place the Kerberos US Feature Into Production	94
6.5.4.1	Copy Kerberos US Feature Files Into Production	94
6.5.4.2	Configure the Kerberos US Feature	96
6.5.4.3	Kerberos US Feature Runtime and Sample Configuration Files	97
6.5.4.4	Unpack Source Files	99
6.6	Installing TCP/IP V2 R4 for VM - Kerberos WT Feature	100
6.6.1	Plan Your Installation for the Kerberos WT Feature	100
6.6.2	Allocate Resources for Installing the Kerberos WT Feature	103
6.6.2.1	PROFILE Directory Entries	104
6.6.2.2	Installing the Kerberos WT Feature to Minidisks	104
6.6.2.3	Installing the Kerberos WT Feature to SFS Directories	105
6.6.3	Install the Kerberos WT Feature	107
6.6.3.1	Update Build Status Table for the Kerberos WT Feature	110
6.6.4	Place the Kerberos WT Feature Into Production	110
6.6.4.1	Copy Kerberos WT Feature Files Into Production	110
6.6.4.2	Configure the Kerberos WT Feature	112
6.6.4.3	Kerberos WT Feature Runtime and Sample Configuration Files	113
6.6.4.4	Unpack Source Files	115
<b>7.0</b>	<b>Service Instructions</b>	<b>116</b>
7.1	VMSES/E Service Process Overview	116
7.2	Servicing TCP/IP V2 R4 for VM	118
7.2.1	Important Service Notes	118
7.2.1.1	PPF Override Considerations	118
7.2.1.2	Language Environment Runtime Library Considerations	118
7.2.2	Preventive Service for TCP/IP V2 R4 for VM	119
7.2.2.1	Prepare to Receive Service	119
7.2.2.2	Receive the Service	121
7.2.2.3	Apply the Service	122
7.2.2.4	Update the Build Status Table	123
7.2.2.5	Build Serviced Objects	125
7.2.2.6	Test the New Service	126
7.2.2.7	Place the Service into Production	126
7.2.2.8	Determine RSU Service Level	126

7.2.3	Corrective Service for TCP/IP V2 R4 for VM	126
7.2.3.1	Prepare to Receive Service	127
7.2.3.2	Receive the Service	128
7.2.3.3	Apply the Service	129
7.2.3.4	Update the Build Status Table	130
7.2.3.5	Build Serviced Objects	132
7.2.3.6	Test the New Service	133
7.2.4	Place the New TCP/IP V2 R4 for VM Service Into Production	133
7.2.4.1	Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production	133
7.2.4.2	Configure TCP/IP V2 R4 for VM	135
7.3	Servicing TCP/IP V2 R4 for VM - Source Feature	137
7.4	Servicing TCP/IP V2 R4 for VM - NFS Feature	138
7.4.1	Important Service Notes	138
7.4.1.1	PPF Override Considerations	138
7.4.1.2	Language Environment Runtime Library Considerations	138
7.4.2	Preventive Service for NFS Feature (RSU)	138
7.4.2.1	Prepare to Receive Service	138
7.4.2.2	Receive the Service	139
7.4.2.3	Apply the Service	140
7.4.3	Update the Build Status Table	141
7.4.3.1	Build Serviced Objects	143
7.4.3.2	Test the New Service	144
7.4.3.3	Place the Service into Production	144
7.4.3.4	Determine RSU Service Level	144
7.4.4	Corrective Service for NFS Feature	144
7.4.4.1	Prepare to Receive Service	144
7.4.4.2	Receive the Service	146
7.4.4.3	Apply the Service	147
7.4.5	Update the Build Status Table	147
7.4.5.1	Build Serviced Objects	150
7.4.5.2	Test the New Service	150
7.4.6	Place the New NFS Feature Service Into Production	150
7.4.6.1	Copy the New NFS Feature Serviced Files Into Production	150
7.4.6.2	Configure the NFS Feature	152
7.5	Servicing TCP/IP V2 R4 for VM - Kerberos US Feature	154
7.5.1	Important Service Notes	154
7.5.1.1	PPF Override Considerations	154
7.5.1.2	Language Environment Runtime Library Considerations	154
7.5.2	Corrective Service for Kerberos US Feature	154
7.5.2.1	Prepare to Receive Service	154
7.5.2.2	Receive the Service	156
7.5.2.3	Apply the Service	156
7.5.3	Update the Build Status Table	157
7.5.3.1	Build Serviced Objects	159
7.5.3.2	Test the New Service	159
7.5.4	Place the New Kerberos US Feature Service Into Production	159

7.5.4.1	Copy the New Kerberos US Feature Serviced Files Into Production	159
7.5.4.2	Configure the Kerberos US Feature	161
7.6	Servicing TCP/IP V2 R4 for VM - Kerberos WT Feature	163
7.6.1	Important Service Notes	163
7.6.1.1	PPF Override Considerations	163
7.6.1.2	Language Environment Runtime Library Considerations	163
7.6.2	Corrective Service for Kerberos WT Feature	163
7.6.2.1	Prepare to Receive Service	163
7.6.2.2	Receive the Service	165
7.6.2.3	Apply the Service	165
7.6.2.4	Update the Build Status Table	166
7.6.2.5	Build Serviced Objects	168
7.6.2.6	Test the New Service	168
7.6.3	Place the New Kerberos WT Feature Service Into Production	168
7.6.3.1	Copy the New Kerberos WT Feature Serviced Files Into Production	168
7.6.3.2	Configure the Kerberos WT Feature	170
<b>Appendix A.</b>	<b>TCP2PROD</b>	172
A.1	TCP2PROD	172
A.1.5	The TCP2PROD \$MSGLOG	173
A.2	TCP/IP for VM CATALOG Files	175
	<b>Appendix B. Modifying TCP/IP for VM CATALOG Files</b>	178
/	<b>Appendix C. Copying TCP/IP for VM Client Code to the Y-Disk</b>	182
	<b>Appendix D. TCP/IP for VM Build Lists</b>	186
D.1	TCP/IP V2 R4 for VM Build Lists	186
D.2	NFS Feature Build Lists	187
D.3	Kerberos US Feature Build Lists	188
D.4	Kerberos WT Feature Build Lists	188
	<b>Appendix E. IBM Network Station Support</b>	190
	E.1 IBM Network Station Support Overview	190
	E.2 IBM Network Station Support Requirements	191
	E.2.1 Operating System Requirements	191
	E.2.2 DASD Storage Requirements	191
	E.2.3 Byte File System (BFS) Storage Requirements	191
	E.2.4 User ID Requirements	192
	E.2.5 Changed User ID Requirements	192
	E.2.5.1 P735FALK Directory Entry Requirements	192
	E.2.5.2 Other P735FALK user ID Requirements	193
	E.2.6 Server Run-time and Configuration Files	193
	E.2.7 Service Considerations	194
	E.2.7.1 Maintaining Multiple NSTATION TARBIN (TARnnnnn) Files	194
	E.2.7.2 Pre-Service Processing	194

	E.2.8 New and Changed Files . . . . .	195
	E.2.8.1 New Files - PQ01770 . . . . .	195
	E.2.8.2 New Files - PQ02301 . . . . .	196
	E.2.8.3 Changed Files - PQ01770 . . . . .	196
	E.2.8.4 Changed Files - PQ04280 . . . . .	196
/	<b>Appendix F. DHCPD Server Support</b> . . . . .	197
/	F.1 DHCPD Server Support Overview . . . . .	197
/	F.2 IBM Network Station Support Requirements . . . . .	197
/	F.2.1 Operating System Requirements . . . . .	198
/	F.2.2 DASD Storage Requirements . . . . .	198
/	F.2.3 User ID Requirements . . . . .	198
/	F.2.4 Changed User ID Requirements . . . . .	198
/	F.2.4.1 P735FALK Directory Entry Requirements . . . . .	198
/	F.2.5 Server Run-time and Configuration Files . . . . .	199
/	F.2.6 New and Changed Files . . . . .	200
/	F.2.6.1 New Files - PQ05305 . . . . .	200
/	F.2.6.2 Changed Files - PQ05305 . . . . .	200
	<b>Appendix G. Overriding the TFTP Minidisk Link Address</b> . . . . .	201
	<b>Appendix H. Configuring the BOOTPD Virtual Machine</b> . . . . .	204
	H.1 Update the TCPIP Server Configuration File . . . . .	204
	H.2 Update the BOOTPD Profile Exit . . . . .	205
	H.3 Configure the ETC BOOTPTAB File . . . . .	205
	H.4 BOOTPD Command . . . . .	206
	H.5 BOOTPD Subcommands . . . . .	209
	H.6 CMS Subcommand . . . . .	210
	H.7 CONFIG Subcommand . . . . .	210
	H.8 EXCLUDE Subcommand . . . . .	214
	H.9 EXIT Subcommand . . . . .	215
	H.10 FORWARD Subcommand . . . . .	215
	H.11 HELP Subcommand . . . . .	219
	H.12 INCLUDE Subcommand . . . . .	219
	H.13 LURK Subcommand . . . . .	220
	H.14 QUIT Subcommand . . . . .	221
	H.15 RELOAD Subcommand . . . . .	221
	H.16 STAYUP Subcommand . . . . .	223
	H.17 STOP Subcommand . . . . .	224
	H.18 TRACE Subcommand . . . . .	225
/	<b>Appendix I. Configuring the DHCPD Virtual Machine</b> . . . . .	226
/	I.1 Update the TCPIP Server Configuration File . . . . .	226
/	I.2 Update the DHCPD Profile Exit . . . . .	227
/	I.3 Configure the ETC DHCPTAB File . . . . .	227
/	I.4 DHCPD Command . . . . .	228



/ I.5 DHCPD Subcommands . . . . .	231
/ I.6 CMS Subcommand . . . . .	232
/ I.7 CONFIG Subcommand . . . . .	233
/ I.8 DELETE Subcommand . . . . .	236
/ I.9 EXCLUDE Subcommand . . . . .	238
/ I.10 EXIT Subcommand . . . . .	239
/ I.11 FORWARD Subcommand . . . . .	240
/ I.12 HELP Subcommand . . . . .	242
/ I.13 INCLUDE Subcommand . . . . .	243
/ I.14 LURK Subcommand . . . . .	244
/ I.15 QUIT Subcommand . . . . .	245
/ I.16 RELOAD Subcommand . . . . .	246
/ I.17 SHOW Subcommand . . . . .	248
/ I.18 STATUS Subcommand . . . . .	252
/ I.19 STAYUP Subcommand . . . . .	255
/ I.20 STOP Subcommand . . . . .	256
/ I.21 TRACE Subcommand . . . . .	256
/ I.22 DHCPD Machine Statements . . . . .	257
/ I.22.1 Machine File Overview . . . . .	257
/ I.23 Statement Precedence Order . . . . .	260
/ I.24 Constructing a DHCPD Machine File . . . . .	261
/ I.24.1 Global Information . . . . .	261
/ I.24.1.1 Ping Time . . . . .	261
/ I.24.1.2 Reserved Address Time . . . . .	262
/ I.24.1.3 Setting the Lease Duration . . . . .	262
/ I.24.1.4 Setting Wait Time Before Reusing an Expired Lease . . . . .	262
/ I.24.1.5 Setting How Often to Check Addresses . . . . .	262
/ I.24.1.6 Supporting BootP Requests . . . . .	263
/ I.24.1.7 Supporting Requests from Unlisted Clients . . . . .	263
/ I.24.1.8 Specifying the Next Server to Use in the Boot Process . . . . .	263
/ I.24.1.9 Defining New Configuration Options . . . . .	263
/ I.24.1.10 Global Options . . . . .	264
/ I.24.1.11 Specifying Option 43 Vendor Data . . . . .	265
/ I.24.1.12 Specifying Global Class Data . . . . .	265
/ I.24.1.13 Specifying Global Client Data . . . . .	266
/ I.24.2 Subnet Related Data . . . . .	267
/ I.25 Machine File Format . . . . .	273
/ I.26 { (Left Brace) Statement . . . . .	275
/ I.27 } (Right Brace) Statement . . . . .	276
/ I.28 Balance: Statement . . . . .	276
/ I.29 BootStrapServer Statement . . . . .	277
/ I.30 CLASS Statement . . . . .	278
/ I.31 CLIENT Statement . . . . .	279
/ I.32 InOrder: Statement . . . . .	281
/ I.33 DefineOptions Statement . . . . .	282
/ I.34 LeaseExpireInterval Statement . . . . .	282

/ I.35 LeaseTimeDefault Statement . . . . .	283
/ I.36 OPTION Statement . . . . .	284
/ I.37 PingTime Statement . . . . .	288
/ I.38 ReservedTime Statement . . . . .	288
/ I.39 SUBNET Statement . . . . .	289
/ I.40 SupportBootP Statement . . . . .	292
/ I.41 SupportUnlistedClients Statement . . . . .	292
/ I.42 UsedIPAddressExpireInterval Statement . . . . .	293
/ I.43 VENDOR Statement . . . . .	294
/ I.44 DHCP Options . . . . .	296
<b>Appendix J. Configuring the TFTP Virtual Machine</b> . . . . .	314
J.1 Update the TCPIP Server Configuration File . . . . .	314
J.2 Update the TFTP Profile Exit . . . . .	315
J.3 Additional TFTP Configuration Considerations . . . . .	315
J.3.1 Restricting TFTP Access to Files . . . . .	315
J.3.2 Changing the Default TFTP BFS Directory . . . . .	316
J.3.3 Controlling TFTP Data Translation . . . . .	316
J.3.4 Collecting TFTP Monitor Data . . . . .	316
J.4 Create the TFTP PERMLIST Data File . . . . .	316
J.5 Create the TFTP USERLIST Data File . . . . .	317
J.6 TFTP Command . . . . .	318
J.7 TFTP Subcommands . . . . .	320
J.8 CACHE Subcommand . . . . .	321
J.9 CLIENTS Subcommand . . . . .	323
J.10 CMS Subcommand . . . . .	324
J.11 CREATION subcommand . . . . .	324
J.12 DROPFILE Subcommand . . . . .	326
J.13 EXIT Subcommand . . . . .	327
J.14 HELP Subcommand . . . . .	327
J.15 LIGHT Subcommand . . . . .	327
J.16 LOADPERM Subcommand . . . . .	328
J.17 LOADUSER Subcommand . . . . .	328
J.18 QUIT Subcommand . . . . .	329
J.19 STAYUP Subcommand . . . . .	329
J.20 STOP Subcommand . . . . .	330
J.21 TRACE Subcommand . . . . .	330
J.22 XFERMODE Subcommand . . . . .	331
J.23 TFTP File Access Control . . . . .	331
<b>Appendix K. Processing the IBM Network Station Client Code</b> . . . . .	333
K.1 IBM Network Station Service Updates . . . . .	333
K.1.1 Service Level Information . . . . .	333
K.1.2 Service Level Information For APAR PQ05644 . . . . .	334
K.1.3 Additional Processing Notes - APAR PQ05644 . . . . .	336
K.1.3.1 Processing Notes for PQ05127 . . . . .	336

	K.1.3.2 Processing Notes for PQ05644	337
	K.1.4 Service Level Information For APAR PQ06451	337
	K.1.5 Service Level Information For APAR PQ08053	338
	K.1.6 Service Level Information For APAR PQ08331	339
	K.1.7 Service Level Information For APAR PQ12439	339
	K.2 IBM Network Station Processing Requirements	340
	K.3 NSTATION LOADBFS Modifications for Installing in a Non-default File Pool	341
	K.4 Process the NSTATION TARBIN File	342
	K.5 IBM Network Station Customization	344
	K.5.1 IBM Network Station Configuration Files	344
	K.5.2 IBM Network Station BOOT Configuration	344
	<b>Appendix L. BOOTPD Trace Records</b>	347
	L.1 BOOTPD Trace Record Format	347
	L.2 BOOTPD Trace Codes	347
/	<b>Appendix M. DHCPD Trace Records</b>	349
/	M.1 DHCPD Trace Record Format	349
/	M.2 DHCPD Trace Codes	349
	<b>Appendix N. TFTPDP Trace Records</b>	352
	N.1 TFTPDP Trace Record Format	352
	N.2 TFTPDP Codes	353
	N.2.1 TFTPDP Trace Entry: 1000	353
	N.2.2 TFTPDP Trace Entry: 1500	354
	N.2.3 TFTPDP Trace Entry: 2000	354
	N.2.4 TFTPDP Trace Entry: 2500	355
	N.2.5 TFTPDP Trace Entry: 3000	355
	N.2.6 TFTPDP Trace Entry: 3500	355
	N.2.7 TFTPDP Trace Entry: 4000	355
	N.2.8 TFTPDP Trace Entry: 4100	356
	N.2.9 TFTPDP Trace Entry: 4200	356
	N.2.10 TFTPDP Trace Entry: 4300	356
	N.2.11 TFTPDP Trace Entry: 5000	356
	N.2.12 TFTPDP Trace Entry: 5100	356
	N.2.13 TFTPDP Trace Entry: 5200	357
	N.2.14 TFTPDP Trace Entry: 6100	357
	N.2.15 TFTPDP Trace Entry: 6200	357
	N.2.16 TFTPDP Trace Entry: 6300	357
	N.2.17 TFTPDP Trace Entry: 6301	358
	N.2.18 TFTPDP Trace Entry: 6302	358
	N.2.19 TFTPDP Trace Entry: 6303	358
	<b>Appendix O. TFPTD APPLDATA Monitor Records</b>	360
	<b>Reader's Comments</b>	362

---

## Figures

1.	Basic Material: Program Tape for TCP/IP V2 R4 for VM	4
2.	Program Tape: File Content for TCP/IP V2 R4 for VM	4
3.	Basic Material: Program Tape for Source Feature	5
4.	Program Tape: File Content for Source Feature	6
5.	Basic Material: Program Tape for NFS Feature	6
6.	Program Tape: File Content for NFS Feature	6
7.	Basic Material: Program Tape for Kerberos US Feature	7
8.	Program Tape: File Content for Kerberos US Feature	7
9.	Basic Material: Program Tape for Kerberos WT Feature	8
10.	Program Tape: File Content for Kerberos WT Feature	8
11.	Basic Material: Unlicensed Publications	9
12.	Optional Material: Licensed Publications	9
13.	Publications Useful During Installation / Service on VM/ESA Version 1	10
14.	Publications Useful During Installation / Service on VM/ESA Version 2	10
15.	PSP Upgrade and Subset ID	11
16.	Component IDs	11
17.	Problem Documentation Addresses	13
18.	Cumulative Service (RSU) Tape Information	15
19.	DASD Storage Requirements for Target Minidisks - Base Product	23
20.	DASD Storage Requirements for Target Minidisks - Source Feature	27
21.	DASD Storage Requirements for Target Minidisks - IBM Network Station Support	28
22.	Additional DASD Storage Requirements for Existing Minidisks - IBM Network Station Support	29
/ 23.	DASD Storage Requirements for Target Minidisks - DHCPD Server Support	30
/ 24.	Additional DASD Storage Requirements for Existing Minidisks - DHCPD Server Support	31
25.	DASD Storage Requirements for Target Minidisks - NFS Feature	32
26.	Additional DASD Storage Requirements for Existing Minidisks - NFS Feature	33
27.	DASD Storage Requirements for Target Minidisks - Kerberos US Feature	34
28.	Additional DASD Storage Requirements for Existing Minidisks - Kerberos US Feature	35
29.	DASD Storage Requirements for Target Minidisks - Kerberos WT Feature	36
30.	Additional DASD Storage Requirements for Existing Minidisks - Kerberos WT Feature	37
31.	Files to Copy into Production - TCP/IP V2 R4 for VM	59
32.	Files to Customize - TCP/IP V2 R4 for VM	60
33.	Files to Copy into Production - NFS Feature	82
34.	Files to Customize - NFS Feature	82
35.	Files to Copy into Production - Kerberos US Feature	97
36.	Files to Customize - Kerberos US Feature	98
37.	Files to Copy into Production - Kerberos WT Feature	113
38.	Files to Customize - Kerberos WT Feature	114
39.	TCP/IP V2 R4 for VM CATALOG Files	175
40.	VMSES/E Build Lists - TCP/IP V2 R4 for VM	186
41.	VMSES/E Build Lists - NFS Feature	187
42.	VMSES/E Build Lists - Kerberos US Feature	188

43.	VMSES/E Build Lists - Kerberos WT Feature	188
44.	Files to Copy into Production - IBM Network Station Support	194
45.	Files to Customize - IBM Network Station Support	194
/ 46.	Files to Copy into Production - DHCPD Server Support	200
/ 47.	Files to Customize - DHCPD Server Support	200
48.	BOOTPD Subcommands	209
/ 49.	DHCPD Subcommands	231
/ 50.	DHCPD Machine File - Example Subnet Environment	267
/ 51.	DHCPD Machine File Statements	274
/ 52.	DHCP Option Names and Numbers	296
/ 53.	DHCP Option Datatype Formats	299
/ 54.	DHCP Options	299
55.	TFTPD Subcommands	320
56.	NSTATION TARBIN Fixes - APAR PQ05644	334
57.	NSTATION TARBIN Fixes - APAR PQ06451	337
58.	NSTATION TARBIN Fixes - APAR PQ08053	338
59.	NSTATION TARBIN Fixes - APAR PQ08331	339
60.	NSTATION TARBIN Fixes - APAR PQ12439	339
61.	IBM Network Station Setup Utility Screen	345
62.	IBM Network Station Set Boot Parameters Screen	346
63.	CP Header Data	360
64.	TFTPD APPLDATA Data	361

---

## Notices

References in this document to IBM\* products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe on any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

International Business Machines Corporation  
IBM Director of Licensing  
500 Columbus Avenue  
Thornwood, NY 10594  
USA

---

## Trademarks and Service Marks

The following terms, denoted by an asterisk (\*) used in this document, are trademarks or service marks of IBM Corporation in the United States or other countries:

IBM  
VM/ESA  
IBMLink (Service Link)

The following terms, denoted by a double asterisk (\*\*) in this document, are trademarks of other companies:

<b>Trademark</b>	<b>Owned By</b>
NCS	Apollo Computer, Inc.
NFS	Sun Microsystems, Inc.
Network File System	Sun Microsystems, Inc.
Portmapper	Sun Microsystems, Inc.

---

## 1.0 Introduction

This program directory is intended for the system programmer responsible for program installation and maintenance. It contains information concerning the material and procedures associated with the installation and service of the following:

- TCP/IP V2 R4 for VM base product
- Source Feature
- NFS Feature
- Kerberos US Feature
- Kerberos WT Feature

You should read all of this program directory before you install this program, then keep this document for future reference.

The program directory contains the following sections:

- 2.0, "Program Materials" on page 4 identifies the basic and optional program materials and documentation for TCP/IP V2 R4 for VM and its corresponding features.
- 3.0, "Program Support" on page 11 describes the IBM support available for TCP/IP V2 R4 for VM and its features.
- 4.0, "Program and Service Level Information" on page 14 lists the APARs (program level) and PTFs (service level) incorporated into TCP/IP V2 R4 for VM and its features.
- 5.0, "Installation Requirements and Considerations" on page 16 identifies the resources and considerations for installing and using TCP/IP V2 R4 for VM and its features.
- 6.0, "Installation Instructions" on page 38 provides detailed installation instructions for TCP/IP V2 R4 for VM and its features. These instructions are comprised of the following:
  - 6.2, "Installing TCP/IP V2 R4 for VM" on page 40
  - 6.3, "Installing TCP/IP V2 R4 for VM - Source Feature" on page 62
  - 6.4, "Installing TCP/IP V2 R4 for VM - NFS Feature" on page 69
  - 6.5, "Installing TCP/IP V2 R4 for VM - Kerberos US Feature" on page 84
  - 6.6, "Installing TCP/IP V2 R4 for VM - Kerberos WT Feature" on page 100
- 7.0, "Service Instructions" on page 116 provides detailed servicing instructions for TCP/IP V2 R4 for VM and its features. These instructions are comprised of the following:
  - 7.2, "Servicing TCP/IP V2 R4 for VM" on page 118
  - 7.4, "Servicing TCP/IP V2 R4 for VM - NFS Feature" on page 138
  - 7.5, "Servicing TCP/IP V2 R4 for VM - Kerberos US Feature" on page 154
  - 7.6, "Servicing TCP/IP V2 R4 for VM - Kerberos WT Feature" on page 163
- Appendix A, "TCP2PROD" on page 172 provides information about the TCP2PROD command, supplied for placing TCP/IP for VM files into production.

- | • Appendix B, “Modifying TCP/IP for VM CATALOG Files” on page 178 provides information about how to change the product CATALOG files via a VMSES/E local modification.
- | • Appendix C, “Copying TCP/IP for VM Client Code to the Y-Disk” on page 182 provides information about copying client files to the system Product Code minidisk.
- | • Appendix D, “TCP/IP for VM Build Lists” on page 186 provides information about the VMSES/E build lists used to maintain TCP/IP for VM.
- | • Appendix E, “IBM Network Station Support” on page 190 provides information about TCP/IP for VM requirements and changes associated with the corrective service which provides this support.
- / • Appendix F, “DHCPD Server Support” on page 197 provides information about TCP/IP for VM requirements and changes associated with the corrective service which provides this support.
- | • Appendix G, “Overriding the TFTPd Minidisk Link Address” on page 201 provides the steps for creating a PPF override that will change the P735FALK link address for the TFTPd 191 minidisk.
- | • Appendix H, “Configuring the BOOTPD Virtual Machine” on page 204 provides setup instructions for the BOOTPD server, and describes the BOOTPD command and subcommands.
- / • Appendix I, “Configuring the DHCPD Virtual Machine” on page 226 provides setup instructions for the DHCPD server, and describes the DHCPD command and subcommands.
- | • Appendix J, “Configuring the TFTPd Virtual Machine” on page 314 provides setup instructions for the TFTPd server, and describes the TFTPd command and subcommands.
- | • Appendix K, “Processing the IBM Network Station Client Code” on page 333 describes how to process the NSTATION TARBIN file, which contains the Client Code files associated with this device.
- | • Appendix L, “BOOTPD Trace Records” on page 347 describes the format of the BOOTPD trace output.
- / • Appendix M, “DHCPD Trace Records” on page 349 describes the format of the DHCPD trace output.
- | • Appendix N, “TFTPd Trace Records” on page 352 describes the format of the TFTPd trace output.
- | • Appendix O, “TFPTD APPLDATA Monitor Records” on page 360 describes the monitor records produced by the TFTPd server.

Before you install TCP/IP V2 R4 for VM, read 3.1, “Preventive Service Planning” on page 11. This section tells you how to find any updates to the information and procedures in this program directory.

---

## 1.1 Program Description

TCP/IP (Transmission Control Protocol/Internet Protocol) enables VM customers to participate in a multivendor, open networking environment using the TCP/IP protocol suite for communications and interoperability. The applications provided in TCP/IP include the ability to transfer files, send mail, log on a remote host, allow access from any other TCP/IP node in the network, and perform other network client and server functions.



IBM® Transmission Control Protocol/Internet Protocol Version 2 Release 4 for VM (TCP/IP V2 R4 for VM) contains the functions provided in TCP/IP for VM Version 2 Release 3, and provides the following enhancements:

- Support of a TELNET Session Connection Exit. This exit allows an initial VM/ESA® CP command (such as the VM/ESA **CP DIAL** command) to be simulated for clients which use a transparent mode terminal.
- Removal of the 2000 TELNET session limit. The number of TELNET sessions is now limited only by the amount of available virtual storage.
- FTP (File Transfer Protocol) server support for the Byte File System (BFS) of VM/ESA (in addition to the current support for the VM/ESA Shared File System (SFS) and VM/ESA minidisks).
- Control over use of TCP/IP services, via the use of TCPIP PERMIT and RESTRICT statements.
- Miscellaneous documentation improvements, which include added documentation about the IUCV Socket interface.
- Miscellaneous service APARs for previous TCP/IP for VM releases have been incorporated in the base code of this release.

---

## 2.0 Program Materials

An IBM program is identified by a program number. The program number for TCP/IP Version 2 Release 4 for VM is 5735-FAL.

The program announcement material describes the features supported by TCP/IP V2 R4 for VM. Ask your IBM marketing representative for this information if you have not already received a copy.

The following sections identify:

- basic and optional program materials available with this program
- publications useful during installation.

---

### 2.1 Basic Machine-Readable Material

The distribution medium for this program is 9-track magnetic tape (written at 6250 BPI), 3480 tape cartridge, or 4-mm tape cartridge. The tape or cartridge contains all the programs and data needed for installation. See 6.0, "Installation Instructions" on page 38 for more information about how to install the program. Figure 1 describes the tape or cartridge. Figure 2 describes the file content of the program tape or cartridge.

*Figure 1. Basic Material: Program Tape for TCP/IP V2 R4 for VM*

Feature Number	Medium	Physical Volume	Tape Content	External Tape Label
5801	6250 tape	1	TCP/IP V2 R4 for VM	TCP/IP VM Base
5801	6250 tape	2	Recommended Service Upgrade	TCP/IP VM <i>yynn</i> RSU
5802	3480 cart.	1	TCP/IP V2 R4 for VM	TCP/IP VM Base
5802	3480 cart.	2	Recommended Service Upgrade	TCP/IP VM <i>yynn</i> RSU
5750	4-mm cart.	1	TCP/IP V2 R4 for VM	TCP/IP VM Base
5750	4-mm cart.	2	Recommended Service Upgrade	TCP/IP VM <i>yynn</i> RSU

**Note:** If a Recommended Service Upgrade (RSU) tape is available, such a tape will accompany each order. This tape will be labeled "yynnRSU," where *yynn* indicates the RSU level.

*Figure 2 (Page 1 of 2). Program Tape: File Content for TCP/IP V2 R4 for VM*

Tape File	Content
1	Tape Header
2	Tape Header

Figure 2 (Page 2 of 2). Program Tape: File Content for TCP/IP V2 R4 for VM

<b>Tape File</b>	<b>Content</b>
3	Product Header
4	Product Memo
5	Service Apply Lists
6	PTFPARTs
7	TCP/IP V2 R4 for VM Service
8	TCP/IP V2 R4 for VM Service
9	TCP/IP V2 R4 for VM Executable Server Code
10	TCP/IP V2 R4 for VM Executable NCS** Code
11	TCP/IP V2 R4 for VM Sample Files
12	TCP/IP V2 R4 for VM Softcopy Publications
13	TCP/IP V2 R4 for VM Base Code
14	TCP/IP V2 R4 for VM Executable Client Code

## 2.2 Optional Machine-Readable Material

There are 4 features for TCP/IP V2 R4 for VM: Source feature, Network File System\*\* (NFS\*\*) feature, Kerberos US feature, and Kerberos World Trade feature.

### 2.2.1 Optional Machine-Readable Material for Source Feature

The distribution medium for this program is 9-track magnetic tape (written at 1600 or 6250 BPI), 3480 tape cartridge, or 4-mm tape cartridge. The tape or cartridge contains all the programs and data needed for installation. See 6.0, "Installation Instructions" on page 38 for more information about how to install the program. Figure 3 describes the tape or cartridge. Figure 4 on page 6 describes the file content of the program tape or cartridge.

Figure 3. Basic Material: Program Tape for Source Feature

<b>Feature Number</b>	<b>Medium</b>	<b>Physical Volume</b>	<b>Tape Content</b>	<b>External Tape Label</b>
5811	6250 tape	1	Source Feature	TCP/IP VM Source
5812	3480 cart.	1	Source Feature	TCP/IP VM Source
5751	4-mm cart.	1	Source Feature	TCP/IP VM Source

Figure 4. Program Tape: File Content for Source Feature

<b>Tape File</b>	<b>Content</b>
1	Tape Header
2	Tape Header
3	Product Header
4	Product Memo
5	Service Apply Lists
6	PTFPARTs
7	Source Feature Service
8	Source Feature Service
9	TCP/IP V2 R4 for VM Source Files

## 2.2.2 Optional Machine-Readable Material for NFS Feature

The distribution medium for this program is 9-track magnetic tape (written at 1600 or 6250 BPI), 3480 tape cartridge, or 4-mm tape cartridge. The tape or cartridge contains all the programs and data needed for installation. See 6.0, "Installation Instructions" on page 38 for more information about how to install the program. Figure 5 describes the tape or cartridge. Figure 6 describes the file content of the program tape or cartridge.

Figure 5. Basic Material: Program Tape for NFS Feature

<b>Feature Number</b>	<b>Medium</b>	<b>Physical Volume</b>	<b>Tape Content</b>	<b>External Tape Label</b>
5821	6250 tape	1	NFS Feature	TCP/IP VM NFS
5822	3480 cart.	1	NFS Feature	TCP/IP VM NFS
5752	4-mm cart.	1	NFS Feature	TCP/IP VM NFS

Figure 6 (Page 1 of 2). Program Tape: File Content for NFS Feature

<b>Tape File</b>	<b>Content</b>
1	Tape Header
2	Tape Header
3	Product Header
4	Product Memo
5	Service Apply Lists
6	PTFPARTs

Figure 6 (Page 2 of 2). Program Tape: File Content for NFS Feature

<b>Tape File</b>	<b>Content</b>
7	NFS Feature Service
8	NFS Feature Service
9	NFS Feature Base Code
10	NFS Feature Executable Server Code
11	NFS Feature Executable Client Code
12	NFS Feature Source Files

### 2.2.3 Optional Machine-Readable Material for Kerberos US Feature

The distribution medium for this program is 9-track magnetic tape (written at 1600 or 6250 BPI), 3480 tape cartridge, or 4-mm tape cartridge. The tape or cartridge contains all the programs and data needed for installation. See 6.0, "Installation Instructions" on page 38 for more information about how to install the program. Figure 7 describes the tape or cartridge. Figure 8 describes the file content of the program tape or cartridge.

Figure 7. Basic Material: Program Tape for Kerberos US Feature

<b>Feature Number</b>	<b>Medium</b>	<b>Physical Volume</b>	<b>Tape Content</b>	<b>External Tape Label</b>
5831	6250 tape	1	Kerberos US Feature	TCP/IP VM Kerb(DES)
5832	3480 cart.	1	Kerberos US Feature	TCP/IP VM Kerb(DES)
5753	4-mm cart.	1	Kerberos US Feature	TCP/IP VM Kerb(DES)

Figure 8 (Page 1 of 2). Program Tape: File Content for Kerberos US Feature

<b>Tape File</b>	<b>Content</b>
1	Tape Header
2	Tape Header
3	Product Header
4	Product Memo
5	Service Apply Lists
6	PTFPARTs
7	Kerberos US Feature Service
8	Kerberos US Feature Service

Figure 8 (Page 2 of 2). Program Tape: File Content for Kerberos US Feature

<b>Tape File</b>	<b>Content</b>
9	Kerberos US Feature Base Code
10	Kerberos US Feature Executable Server Code
11	Kerberos US Feature Executable Client Code
12	Kerberos US Feature Source Files

## 2.2.4 Optional Machine-Readable Material for Kerberos WT Feature

The distribution medium for this program is 9-track magnetic tape (written at 1600 or 6250 BPI), 3480 tape cartridge, or 4-mm tape cartridge. The tape or cartridge contains all the programs and data needed for installation. See 6.0, "Installation Instructions" on page 38 for more information about how to install the program. Figure 9 describes the tape or cartridge. Figure 10 describes the file content of the program tape or cartridge.

Figure 9. Basic Material: Program Tape for Kerberos WT Feature

<b>Feature Number</b>	<b>Medium</b>	<b>Physical Volume</b>	<b>Tape Content</b>	<b>External Tape Label</b>
5841	6250 tape	1	Kerberos WT Feature	TCP/IP Kerb(nonDES)
5842	3480 cart.	1	Kerberos WT Feature	TCP/IP Kerb(nonDES)
5754	4-mm cart.	1	Kerberos WT Feature	TCP/IP Kerb(nonDES)

Figure 10 (Page 1 of 2). Program Tape: File Content for Kerberos WT Feature

<b>Tape File</b>	<b>Content</b>
1	Tape Header
2	Tape Header
3	Product Header
4	Product Memo
5	Service Apply Lists
6	PTFPARTs
7	Kerberos WT Feature Service
8	Kerberos WT Feature Service
9	Kerberos WT Feature Base Code
10	Kerberos WT Feature Executable Server Code

Figure 10 (Page 2 of 2). Program Tape: File Content for Kerberos WT Feature

---

<b>Tape File</b>	<b>Content</b>
11	Kerberos WT Feature Executable Client Code
12	Kerberos WT Feature Source Files

---

## 2.3 Program Publications

The following sections identify the basic and optional publications for TCP/IP V2 R4 for VM. There are no additional publications for the features.

### 2.3.1 Basic Program Publications

One copy of each of the following publications is included when you order the basic materials for TCP/IP V2 R4 for VM. For additional copies, contact your IBM representative.

Figure 11. Basic Material: Unlicensed Publications

---

<b>Publication Title</b>	<b>Form Number</b>
TCP/IP V2 R4 for VM: User's Guide	SC31-6081
TCP/IP V2 R4 for VM: Planning and Customization	SC31-6082
TCP/IP V2 R4 for VM: Programmer's Reference	SC31-6084
TCP/IP V2 R4 for VM: Messages and Codes	SC31-6151
TCP/IP V2 R4 for VM: LPS	GC31-6083

---

### 2.3.2 Optional Program Publications

The following publications can be ordered. Contact your IBM representative.

Figure 12. Optional Material: Licensed Publications

---

<b>Publication Title</b>	<b>Form Number</b>
TCP/IP for VM and MVS Diagnosis Guide	LY43-0013

---

### 2.3.3 Softcopy Publications

The TCP/IP V2 R4 for VM basic program publications, except for the LPS document, are shipped in softcopy format on the TCP/IP V2 R4 for VM installation tape. The product installation procedure includes an optional step for installing the softcopy publications.

---

## 2.4 Microfiche Support

There is no microfiche for TCP/IP V2 R4 for VM.

---

## 2.5 Publications Useful During Installation and Service

The publications listed in Figure 13 may be useful during the installation and service of TCP/IP V2 R4 for VM and its corresponding features. To order copies, contact your IBM representative.

*Figure 13. Publications Useful During Installation / Service on VM/ESA Version 1*

<b>Publication Title</b>	<b>Form Number</b>
VMSES/E Introduction and Reference	SC24-5444
VM/ESA Planning and Administration	SC24-5521
VM/ESA Service Guide	SC24-5527
VM/ESA CMS Command Reference	SC24-5461
VM/ESA SFS and CRR Planning, Administration and Operation	SC24-5649
VM/ESA System Messages and Codes	SC24-5529
TCP/IP V2 R4 for VM: Planning and Customization	SC31-6082
TCP/IP V2 R4 for VM: User's Guide	SC31-6081

*Figure 14. Publications Useful During Installation / Service on VM/ESA Version 2*

<b>Publication Title</b>	<b>Form Number</b>
VMSES/E Introduction and Reference	SC24-5747
VM/ESA Service Guide	SC24-5749
VM/ESA Planning and Administration	SC24-5750
VM/ESA CMS Command Reference	SC24-5776
VM/ESA File Pool Planning, Administration and Operation	SC24-5751
VM/ESA System Messages and Codes	SC24-5784
TCP/IP V2 R4 for VM: Planning and Customization	SC31-6082
TCP/IP V2 R4 for VM: User's Guide	SC31-6081



---

## 3.0 Program Support

This section describes the IBM support available for TCP/IP V2 R4 for VM.

---

### 3.1 Preventive Service Planning

Before you install TCP/IP V2 R4 for VM or any TCP/IP for VM features, check with your IBM Support Center or use IBMLink<sup>®</sup> (ServiceLink) to see if additional Preventive Service Planning (PSP) information is available that you should know. To obtain this information, specify the following UPGRADE and SUBSET values:

Figure 15. PSP Upgrade and Subset ID

---

Retain				
COMPID	Release	Upgrade	Subset	Component Name
5735FAL00	240	TCPIP240	VM240	TCP/IP V2 R4 for VM
5735FAL00	241	TCPIP240	VM240	Kerberos US Feature
5735FAL00	242	TCPIP240	VM240	Kerberos WT Feature
5735FAL00	243	TCPIP240	VM240	NFS Feature
5735FAL00	244	TCPIP240	VM240	Source Feature
5735FAL00	240	TCPIP240	yynnRSU	RSU Service Recommendations
5735FAL00	240	TCPIP240	RSU-BY-PTF	Sorted by PTF
5735FAL00	240	TCPIP240	RSU-BY APAR	Sorted by APAR
5735FAL00	240	TCPIP240	RSU-BY-LEVEL	Sorted by RSU level

---

### 3.2 Statement of Support Procedures

Report any difficulties you have using this program to your IBM Support Center. If an APAR (or, Authorized Program Analysis Report) is required, the Support Center will provide the address to which any needed documentation can be sent.

Figure 16 identifies the component ID (COMPID), Retain Release and Field Engineering Service Number (FESN) for TCP/IP V2 R4 for VM and the features.

Figure 16 (Page 1 of 2). Component IDs

---

Retain			
COMPID	Release	Component Name	FESN
5735FAL00	240	TCP/IP V2 R4 for VM	0461035

---

Figure 16 (Page 2 of 2). Component IDs

---

Retain			
COMPID	Release	Component Name	FESN
5735FAL00	241	Kerberos US Feature	0461035
5735FAL00	242	Kerberos WT Feature	0461035
5735FAL00	243	NFS Feature	0461035
5735FAL00	244	Source Feature	0461035

---

### 3.3 Service Information

The IBM Software Support Center provides you with telephone assistance in problem diagnosis and resolution. You can call the IBM Software Support Center at any time; you will receive a return call within eight business hours (Monday—Friday, 8:00 a.m.—5:00 p.m., local customer time). The number to call is:

**1-800-237-5511**

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized supplier.

| Some installation and service-related information, such as the Preventive Service Planning (PSP) “bucket,”  
| is also available on-line, via the TCP/IP for VM home page on the World Wide Web. The URL for this  
| home page is:

/ <http://www.ibm.com/s390/vm/related/tcpip/>

#### 3.3.1 Problem Documentation

When working with TCP/IP for VM support personnel on problems associated with an active Problem Management Record (PMR), diagnostic information may be requested at times. In such cases, the support staff will work with you to determine how to provide the requested documentation. For reference, the addresses that can be used to submit various documentation formats are listed in this section.

**Note:** The addresses that follow cannot be used as a problem reporting facility. All requests for problem assistance must be processed through the IBM Software Support Center, as described above. Documentation submitted to any of these addresses will be reviewed only if it is associated with an active PMR.

Figure 17. Problem Documentation Addresses

---

Format	Address
Internet	vmtcpdoc@vnet.ibm.com
IBM Mail	USIB6AZU @ IBMMAIL
IBMLink	GDLVM7(TCPLVL2)
Carrier Service	IBM Corporation Attention: <i>IBM contact name</i> Dept. G79G 1701 North St. Endicott, NY 13760

---

### 3.3.2 Communicating Your Comments to IBM

If you have comments about or suggestions for the TCP/IP V2 R4 for VM program product, or this Program Directory, you can provide them to IBM through one of the following:

- If you prefer to send comments by mail, use the address provided with the Reader's Comments form (RCF) at the back of this document.
- If you prefer to send comments electronically, use this Internet e-mail ID:

**vmtcpip@vnet.ibm.com**

When sending documentation-related comments, please include:

- the title of this publication
- the section title, section number, or topic to which your comment applies.

---

## 4.0 Program and Service Level Information

This section identifies the program and any relevant service levels of TCP/IP V2 R4 for VM and its corresponding features. The program level refers to the APAR fixes incorporated into the program. The service level refers to the PTFs shipped with this product. Information about the cumulative service tape is also provided.

---

### 4.1 Program Level Information - TCP/IP V2 R4 for VM

The following APAR fixes against the previous release of TCP/IP for VM and the Source Feature have been incorporated into this release.

PN46080	PN47781	PN55816	PN57765	PN59425	PN59577
PN60353	PN60371	PN60563	PN60578	PN60736	PN60800
PN61715	PN61812	PN62010	PN62997	PN63187	PN63213
PN63223	PN63303	PN63450	PN63497	PN64069	PN64568
PN64731	PN64796	PN65208	PN65410	PN65563	PN65756
PN65797	PN65986	PN66034	PN66106	PN66340	PN66453
PN66607	PN66652	PN66801	PN66840	PN66864	PN66893
PN66917	PN66977	PN67017	PN67073	PN67240	PN67295
PN67301	PN67360	PN67615	PN67657	PN67733	PN67743
PN67784	PN67894	PN67910	PN67941	PN68137	PN68160
PN68379	PN68783	PN68799	PN68846	PN68945	PN68970
PN69020	PN69136	PN69236	PN69269	PN69385	PN69443
PN69606	PN69827	PN69939	PN69966	PN70508	PN70619
PN71174	PN71368	PN71675	PN72008	PN72460	PN72517
PN72619	PN73210	PN73224	PN73396	PN73740	PN73907
PN73969	PN74045	PN74172	PN74222	PN74269	PN74295
PN74624	PN74751	PN74873	PN75144	PN75145	PN75563
PN75578	PN75752	PN75864	PN75941	PN76063	PN76074
PN76299	PN76462	PN76752	PN76967	PN76976	PN76980
PN77126	PN77290	PN77780	PN77929	PN78111	PN78123
PN78210	PN78363	PN78573	PN78579	PN78637	PN79038
PN79125	PN79883	PN79983	PN80549	PN80954	PN81018
PN81157	PN81207	PN81396	PN81397	PN81449	PN82243
PN83183	PN83532	PN83534	PN83887	PN84000	PN84501
PN84675	PN85090	PN86429	PN86529	PN86620	PN88402

---

## 4.2 Program Level Information - NFS Feature

The following APAR fixes against the previous release of the NFS Feature have been incorporated into this release.

PN66652 PN66848 PN68859

---

## 4.3 Program Level Information - Kerberos US Feature

The following APAR fixes against the previous release of the Kerberos US Feature have been incorporated into this release.

PN47781 PN66652 PN82615

---

## 4.4 Program Level Information - Kerberos WT Feature

The following APAR fixes against the previous release of the Kerberos WT Feature have been incorporated into this release.

PN47781 PN66652 PN82615

---

## 4.5 Service Level Information

Check the TCP/IP240 PSP bucket for any additional PTFs that should be installed or any additional install information you should be aware of before you install the TCP/IP V2 R4 for VM, or any of its corresponding features.

---

## 4.6 Cumulative Service (RSU) Tape Information

Cumulative service for the TCP/IP V2 R4 for VM is available through a periodic preventive service tape, the Recommended Service Upgrade (RSU). The current level of the TCP/IP for VM RSU can be obtained using the information provided in Figure 18:

*Figure 18. Cumulative Service (RSU) Tape Information*

---

Retain			
COMPID	Release	APAR Number	PTF
5735FAL00	RSU	PN92440	UN99240

---

## 5.0 Installation Requirements and Considerations

The following sections identify system requirements for installing and activating TCP/IP V2 R4 for VM, and its corresponding features.

---

### 5.1 Hardware Requirements

There are no special hardware requirements to install TCP/IP V2 R4 for VM or its features. Additional hardware requirements for exploiting specific functions of TCP/IP V2 R4 for VM and its features are documented in the announcement material and in *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082).

---

### 5.2 Program Considerations

The following sections list the programming considerations for installing TCP/IP V2 R4 for VM and its features, and activating their functions.

#### 5.2.1 Operating System Requirements

- VM/ESA Version 2 Release 2.0
- VM/ESA Version 2 Release 1.0
- VM/ESA Version 1 Release 2.2
- Required VMSES/E service
  - **RSU9602 service level** or above must be applied to VMSES/E on VM/ESA Version 1 Release 2.2 prior to installing TCP/IP V2 R4 for VM  
VMSES/E APAR VM59896 (Initial PTF: UM27492) must also be applied, via corrective service.
  - **RSU9601 service level** or above must be applied to VMSES/E on VM/ESA Version 2 Release 1.0 prior to installing TCP/IP V2 R4 for VM
- Required CMS service
  - The following APAR must be applied to CMS on VM/ESA Version 2 Release 1.0 if FTP Server support for the Byte File System (BFS) is used.  
VM60747 (Initial PTF: UM28185)
  - If you plan to install TCP/IP V2 R4 for VM support for the IBM Network Station, the following CMS service must be installed:
    - For VM/ESA Version 2 Release 1.0:  
VM61080 (Initial PTF: UM28330)

|           - For VM/ESA Version 2 Release 2.0:  
|            VM61080 (Initial PTF: UM28331)  
|        See Appendix E, “IBM Network Station Support” on page 190 for more information about these  
|        and other IBM Network Station requirements.  
/       - If you plan to install TCP/IP V2 R4 for VM DHCPD server support, the following CMS service must  
/       be installed:  
/           - For VM/ESA Version 2 Release 1.0:  
/            VM61480 (Initial PTF: UM28656)  
/           - For VM/ESA Version 2 Release 2.0:  
/            VM61480 (Initial PTF: UM28657)  
/        See Appendix F, “DHCPD Server Support” on page 197 for more information about these and  
/        other DHCPD server support requirements.

## 5.2.2 Other Program Product Requirements

### 5.2.2.1 Other Program Product Requirements - TCP/IP V2 R4 for VM

IBM C for VM/ESA Version 3 Release 1 (5654-033) has been used to build the C components that provide the following TCP/IP V2 R4 for VM services. To use these services, the Language Environment runtime library must be available:

- Domain Name Server virtual machine (NAMESRV)
- NCS Local and Global Location Broker daemons (NCS, NCSLLBD, and NCSGLBD)
- Network Data Base servers (NDBPMGR and NDBSRV $nn$ )
- Portmapper\*\* server (PORTMAP)
- Remote Execution daemon (REXECD and RSLAVE $n$ )
- RouteD server (ROUTED)
- SNMP Query Engine and Agent (SNMPD and SNMPQE)
- Sockets Applications Programming Interface

Various client functions require Language Environment runtime library support as well. Representative of these are:

- CMSRESOL and CMSRESXA
- CPP
- DIG
- NSLOOKUP
- RPCGEN and RPCINFO
- UUID\_GEN
- NDBSRVS

The Language Environment runtime library necessary to use these C components — the Common Execution Library (CEL) — is supplied with VM/ESA Version 2. If you are installing TCP/IP V2 R4 for VM

on VM/ESA Version 1 Release 2.2, IBM Language Environment for MVS and VM Release 5 (5688198) must be separately installed to use and service these components.

**Note:** The code for the above services is installed when you install TCP/IP V2 R4 for VM, regardless of whether or not you intend to use all, or only a subset, of these services.

Additional software requirements for exploiting specific functions of TCP/IP V2 R4 for VM are documented in the announcement material and in *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082).

### 5.2.2.2 Other Program Product Requirements - Features

If you are installing TCP/IP V2 R4 for VM on VM/ESA Version 1 Release 2.2, IBM Language Environment for MVS and VM Release 5 (5688198) is required to use the following features:

- NFS Feature
- Kerberos US Feature
- Kerberos WT Feature

The Language Environment runtime library necessary to use the C components of these TCP/IP V2 R4 for VM features. — the Common Execution Library (CEL) — is supplied with VM/ESA Version 2. If you are installing TCP/IP V2 R4 for VM on VM/ESA Version 1 Release 2.2, IBM Language Environment for MVS and VM Release 5 (5688198) must be separately installed to use and service these components.

Additional software requirements for exploiting specific functions of TCP/IP V2 R4 for VM are documented in the announcement material and in *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082).

### 5.2.3 Program Installation/Service Considerations

This section describes items that should be considered before you install or service TCP/IP V2 R4 for VM and its features.

- VMSES/E is required to install and service this product.
- If multiple users install and maintain licensed products on your system, you may encounter problems getting the necessary access to the MAINT 51D minidisk. If you find there is contention for write access to the 51D minidisk, you can eliminate it by converting the Software Inventory from minidisk to Shared File System (SFS). See the *VMSES/E Introduction and Reference*, Chapter 18 “Changing the Software Inventory to an SFS Directory,” for information about how to make this change.
- The **P735FALK** user ID is used to install and service TCP/IP V2 R4 for VM and its corresponding features. This is the IBM suggested user ID name. You are free to change this to any user ID name you wish; however, a PPF override to implement such a change must be created for TCP/IP V2 R4 for VM and each additional TCP/IP feature that you install. feature that you install.
- TCP/IP V2 R4 for VM, and its corresponding features, each have several defined components. The component you need to use for installation and service depends on whether you install to minidisks or the Shared File System, and whether you have installed Language Environment runtime library support and wish to use any of the C components that require this support.



**Note:** Certain minidisks **must** be defined and used with TCP/IP server machines, even when TCP/IP for VM is installed to Shared File System directories. This requirement is explained further in item 4 of 5.3, “DASD Storage and User ID Requirements” on page 20.

The components for TCP/IP V2 R4 for VM are:

- **TCPIP** - install and service the TCP/IP for VM using minidisks
- **TCPIPSFS** - install and service the TCP/IP for VM using the Shared File System
- **TCPIPNOC** - install TCP/IP for VM using minidisks, but only service code that does not require Language Environment runtime library support
- **TCPIPSFSNOC** - install TCP/IP for VM using the Shared File System, but only service code that does not require Language Environment runtime library support

The components for the Source Feature are:

- **SOURCE** - if TCP/IP for VM is installed and serviced using minidisks
- **SOURCESFS** - if TCP/IP for VM is installed and serviced using the Shared File System

The components for the NFS Feature are:

- **VMNFS** - if TCP/IP for VM is installed and serviced using minidisks
- **VMNFSSFS** - if TCP/IP for VM is installed and serviced using the Shared File System

The components for the Kerberos US Feature are:

- **VMKERB** - if TCP/IP for VM is installed and serviced using minidisks
- **VMKERBSFS** - if TCP/IP for VM is installed and serviced using the Shared File System

The components for the Kerberos WT Feature are:

- **VMKERB** - if TCP/IP for VM is installed and serviced using minidisks
- **VMKERBSFS** - if TCP/IP for VM is installed and serviced using the Shared File System
- During the installation and service processes, you may need to perform additional steps to account for errors that cannot be corrected through the service process. If such steps are required, additional notes and text which explain those steps will be included in updated levels of this document.

Several additional considerations for the various TCP/IP for VM features follow:

- You **must** install the base product, TCP/IP V2 R4 for VM, before you install any of the features.
- Install all features using the same installation user ID used to install TCP/IP V2 R4 for VM.
- The NFS Feature, Kerberos US Feature, and Kerberos WT Feature use the same service minidisks and SFS directories as TCP/IP V2 R4 for VM, **except** for APPLY minidisks; each feature has its own set of APPLY minidisks.
- The Source Feature is not separately serviced; its service is included in the service for TCP/IP V2 R4 for VM.

- The source files shipped with each feature are in packed format. You will need to unpack these files before you can use them.

---

## 5.3 DASD Storage and User ID Requirements

The tables provided in this section list the user IDs and minidisks that are used to install and service TCP/IP V2 R4 for VM, and its corresponding features.

### Important Installation Notes:

1. User IDs and minidisks will be defined in
  - 6.2.1, “Plan Your Installation For TCP/IP V2 R4 for VM” on page 40
  - 6.3.1, “Plan Your Installation for the Source Feature” on page 62
  - 6.4.1, “Plan Your Installation for the NFS Feature” on page 69
  - 6.5.1, “Plan Your Installation for the Kerberos US Feature” on page 84
  - 6.6.1, “Plan Your Installation for the Kerberos WT Feature” on page 100

These resources are identified in the tables that follow so you are aware of the resources you'll need prior to allocating them.

2. P735FALK is the default installation and service user ID, but this can be changed. If you choose to change the name of the installation user ID you need to create a Product Parameter (PPF) override. This can be done as TCP/IP for VM and each feature are installed, during Step 6 of the applicable “Plan Your Installation” step

**Note:** If you change the TCP/IP for VM installation user ID to a common or existing user ID (such as MAINT), minidisks may already be defined with addresses which are identical to the TCP/IP for VM minidisk defaults. If any minidisks are in use that have such matching addresses, you will need to create a PPF override to change the default TCP/IP for VM minidisk addresses so they are unique within your environment.

3. The P735FALK, TCPMAINT, and TCP/IP user IDs and minidisks must be defined and installed. The remaining user IDs listed in the tables that follow are for servers and clients associated with TCP/IP V2 R4 for VM optional services. If you choose to not use a particular optional service, you don't need to define the user IDs and production minidisks associated with that service.
4. In the tables that follow, if an SFS 4K block value and default directory name are not provided for a given minidisk, minidisk space **must** be used; such minidisks **cannot** be replaced with an SFS directory.
5. **All** P735FALK test build minidisks **must** be defined. If the P735FALK minidisks for optional services are not defined, problems will be encountered during installation and service.

For example, if you won't be using the NCS Location Broker daemons, you don't need to define the NCS, NCSGLBD and NCSLLBD 191 minidisks; likewise for the NCS 195 minidisk. However, you must still define the P735FALK 395 test build minidisk.

To determine which services (if any) you do not want to use, review the descriptions of services provided in *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082). The server-specific configuration chapters of this publication may also help you with your determination.

Also, if you omit resources that correspond to an optional service, you should not attempt to place the associated files into production, such as during step 6.2.5.1, “Copy TCP/IP V2 R4 for VM Files Into Production” on page 55. The 5735FALK CATALOG file should be modified so that files not used in your environment are bypassed during TCP2PROD command processing. See Appendix A, “TCP2PROD” on page 172 for information about the TCP2PROD command and TCP/IP for VM catalog files.

6. Additional storage may need to be allocated for a given server 191 minidisk, depending on your environment. Some examples of disks that may need to be increased, and reasons for this, follow:

- SMTP 191 - to process a high volume of e-mail
- VMNFS 191 - to support a large number of clients
- DHCPD 191 - to support a large number of clients
- LPSEVER 191 - to allow processing of sizeable print jobs.

Minidisks may also need to be increased to accommodate the logging of trace activity.

7. The storage requirements for some minidisks may be revised over time, which may require existing minidisk sizes to be increased. Any such changes will be indicated in updated levels of this document by revision characters (usually a vertical bar — “|”) in the left margin for the tables in 5.3.1, “DASD Requirements for TCP/IP V2 R4 for VM” on page 23. If the requirements for a given minidisk have changed, use the updated values shown in these tables, instead of those cited in the “PLANINFO” files that are created when TCP/IP for VM or its features are installed.
8. You can install multiple virtual machines, named NDBSRV01, NDBSRV02, etc., if you choose to run the Network Database. Each NDBSRV $n$  virtual machine should be defined similar to NDBSRV01.
9. You can install multiple virtual machines, named RSLAVE1, RSLAVE2, etc., if you choose to run the remote execution daemon (REXECD). Each RSLAVE $n$  virtual machine should be defined similar to RSLAVE1. However, the RSLAVE $n$  virtual machines do not own any minidisks.
10. TCP/IP for VM features use the same service minidisks as TCP/IP V2 R4 for VM, **except** APPLY minidisks. Use the tables that correspond to each feature you plan to install to determine:
  - the storage requirements for any additional service minidisks.
  - additional storage that you need to add to existing service minidisks, if sufficient storage was not already allocated when the base product was installed.

Service minidisks that do not require additional storage are not listed in the product feature tables.

11. Each TCP/IP for VM feature loads files to the TCP/IP for VM Source minidisk. The source storage requirements listed in the program feature tables should be used to increase the size of existing minidisks, or to create a new disk (if the required Source disk does not already exist).
12. Source files are shipped in packed format. If you intend to unpack the source files after installation, ensure you define a sufficiently large Source minidisk or directory, as specified by the disk size for unpacked files in the tables that follow.

13. For information about copying client code to the Product Code minidisk, see Appendix C, “Copying TCP/IP for VM Client Code to the Y-Disk” on page 182.
- | 14. The storage requirements listed in Figure 21 on page 28 and Figure 22 on page 29 are applicable  
| only if you intend to install the corrective service for IBM Network Station support. See Appendix E,  
| “IBM Network Station Support” on page 190 for more information about these and other IBM Network  
| Station requirements.
- / 15. The storage requirements listed in Figure 23 on page 29 and Figure 24 on page 31 are applicable  
/ only if you intend to install the corrective service for DHCPD server support. See Appendix F,  
/ “DHCPD Server Support” on page 197 for more information about these and other DHCPD server  
/ support requirements.

### 5.3.1 DASD Requirements for TCP/IP V2 R4 for VM

Figure 19 (Page 1 of 4). DASD Storage Requirements for Target Minidisks - Base Product

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	191	3390 3380 9345	17 20 20	24000	3000	P735FALK user ID 191 minidisk. <b>VMSYS:P735FALK.</b>
P735FALK	2B2	3390 3380 9345	57 68 68	81600	10200	Contains all the base code shipped with TCP/IP V2 R4 for VM. <b>VMSYS:P735FALK.TCPIP.OBJECT</b>
P735FALK	2B3	3390 3380 9345	0 0 0	0	0	Source code disk. There is no source code on this tape. This disk will need to be added if any of the features is installed. <b>VMSYS:P735FALK.TCPIP.SOURCE</b>
P735FALK	2B4	3390 3380 9345	7 8 8	9600	1200	Contains softcopy publications. This disk can be removed if the publications are not installed. <b>VMSYS:P735FALK.TCPIP.PUBS</b>
P735FALK	2C2	3390 3380 9345	16 19 19	22800	2850	Contains sample files. This disk can be removed if the sample files are not installed. <b>VMSYS:P735FALK.TCPIP.SAMPLE</b>
P735FALK	2C4	3390 3380 9345	5 5 5	6000	750	Contains local modifications. <b>VMSYS:P735FALK.TCPIP.LOCAL</b>
P735FALK	2D2	3390 3380 9345	17 20 20	24000	3000	Contains serviced files. <b>VMSYS:P735FALK.TCPIP.DELTA</b>
P735FALK	2A6	3390 3380 9345	9 10 10	12000	1500	Contains AUX files and software inventory tables that represent the test service level of TCP/IP V2 R4 for VM. <b>VMSYS:P735FALK.TCPIP.APPLYALT</b>

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.
2. For installation to SFS directories, a total of 24000 4K blocks are required. This total accounts for sample files and softcopy pubs; if sample files and softcopy pubs are not installed, 19950 4K blocks are required.
3. See Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182 for information about copying client code to the Product Code minidisk.

Figure 19 (Page 2 of 4). DASD Storage Requirements for Target Minidisks - Base Product

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	2A2	3390 3380 9345	9 10 10	12000	1500	Contains AUX files and software inventory tables that represent the service level of TCP/IP V2 R4 for VM that is currently in production. <b>VMSYS:P735FALK.TCPIP.APPLYPROD</b>
P735FALK	491	3390 3380 9345	23 27 27	32400	_____	Test build disk for server code. This code will be copied to a production disk (TCPMAINT 591), so that disk will also require this amount of free space.
P735FALK	492	3390 3380 9345	50 60 60	72000	_____	Test build disk for client code. This code will be copied to a production disk (TCPMAINT 592), so that disk will also require this amount of free space.
P735FALK	395	3390 3380 9345	2 2 2	2400	_____	Test build disk for NCS code. This code will be copied to a production disk (NCS 195), so that disk will also require this amount of free space.
TCPMAINT	191	3390 3380 9345	7 8 8	9600	_____	TCPMAINT user ID 191 minidisk.
TCPMAINT	198	3390 3380 9345	9 10 10	12000	_____	Contains configuration files for clients and servers.
TCPMAINT	591	3390 3380 9345	23 27 27	32400	_____	Production build disk for server code.
TCPMAINT	592	3390 3380 9345	50 60 60	72000	_____	Production build disk for client code. (2*)

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.
2. For installation to SFS directories, a total of 24000 4K blocks are required. This total accounts for sample files and softcopy pubs; if sample files and softcopy pubs are not installed, 19950 4K blocks are required.
3. See Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182 for information about copying client code to the Product Code minidisk.

Figure 19 (Page 3 of 4). DASD Storage Requirements for Target Minidisks - Base Product

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
FTPSERVE	191	3390 3380 9345	9 10 10	12000	_____	FTPSERVE user ID 191 minidisk.
LPSERVE	191	3390 3380 9345	2 2 2	2400	_____	LPSERVE user ID 191 minidisk.
NAMESRV	191	3390 3380 9345	2 2 2	2400	_____	NAMESRV user ID 191 minidisk.
NCS	191	3390 3380 9345	2 2 2	2400	_____	NCS user ID 191 minidisk.
NCS	195	3390 3380 9345	2 2 2	2400	_____	Production build disk for NCS administration code.
NCSGLBD	191	3390 3380 9345	2 2 2	2400	_____	NCSGLBD user ID 191 minidisk.
NCSLLBD	191	3390 3380 9345	2 2 2	2400	_____	NCSLLBD user ID 191 minidisk.
NDBPMGR	191	3390 3380 9345	1 1 1	1200	_____	NDBPMGR user ID 191 minidisk.
NDBSRV01	191	3390 3380 9345	1 1 1	1200	_____	NDBSRV01 user ID 191 minidisk.
PORTMAP	191	3390 3380 9345	2 2 2	2400	_____	PORTMAP user ID 191 minidisk.

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.
2. For installation to SFS directories, a total of 24000 4K blocks are required. This total accounts for sample files and softcopy pubs; if sample files and softcopy pubs are not installed, 19950 4K blocks are required.
3. See Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182 for information about copying client code to the Product Code minidisk.

Figure 19 (Page 4 of 4). DASD Storage Requirements for Target Minidisks - Base Product

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
REXECD	191	3390 3380 9345	2 2 2	2400	_____	REXECD user ID 191 minidisk.
ROUTED	191	3390 3380 9345	2 2 2	2400	_____	ROUTED user ID 191 minidisk.
RSLAVE1	_____	3390 3380 9345	— — —	_____	_____	REXEC Slave - no 191 minidisk is required; the REXECD 191 minidisk is utilized.
SMTP	191	3390 3380 9345	25 30 30	36000	_____	SMTP user ID 191 minidisk.
SNALNKA	191	3390 3380 9345	3 3 3	3600	_____	SNALNKA user ID 191 minidisk.
SNMPD	191	3390 3380 9345	2 2 2	2400	_____	SNMPD user ID 191 minidisk.
SNMPQE	191	3390 3380 9345	2 2 2	2400	_____	SNMPQE user ID 191 minidisk.
TCPIP	191	3390 3380 9345	5 5 5	6000	_____	TCPIP user ID 191 minidisk.
X25IPI	191	3390 3380 9345	2 2 2	2400	_____	X25IPI user ID 191 minidisk.

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.
2. For installation to SFS directories, a total of 24000 4K blocks are required. This total accounts for sample files and softcopy pubs; if sample files and softcopy pubs are not installed, 19950 4K blocks are required.
3. See Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182 for information about copying client code to the Product Code minidisk.



### 5.3.2 DASD Requirements for Source Feature

<i>Figure 20. DASD Storage Requirements for Target Minidisks - Source Feature</i>						
Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	2B3	3390 3380 9345	33 40 40	48000	6000	Source files disk - in <b>PACKED</b> format <b>VMSYS:P735FALK.TCPIP.SOURCE</b>
P735FALK	2B3	3390 3380 9345	63 75 75	90000	11250	Source files disk - in <b>UNPACKED</b> format <b>VMSYS:P735FALK.TCPIP.SOURCE</b>

**Note:** Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.

For installation to SFS directories, a total of 6000 4K blocks are required. This total accounts for installation of packed source files. If source files will be unpacked, 11250 4K blocks are required.

### 5.3.3 DASD Storage Requirements for IBM Network Station Support

The storage requirements listed in Figure 21 and Figure 22 on page 29 are applicable only if you intend to install the corrective service for IBM Network Station support. See Appendix E, "IBM Network Station Support" on page 190 for more information about these and other IBM Network Station requirements.

#### 5.3.3.1 DASD Storage Requirements for Target Minidisks

Figure 21. DASD Storage Requirements for Target Minidisks - IBM Network Station Support

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	493	3390 3380 9345	64 76 76	91200	11400	Test build disk for IBM Network Station Client Code <b>VMSYS:P735FALK.TCPIP.BUILDNWS</b>
BOOTPD	191	3390 3380 9345	2 2 2	2400	_____	BOOTPD user ID 191 minidisk. (2*)
TFTPD	191	3390 3380 9345	2 2 2	2400	_____	TFTPD user ID 191 minidisk.

**Notes:**

- Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.  
  
11400 4K blocks are needed for SFS install, including the IBM Network Station Client Code (10 4K blocks without the IBM Network Station Client Code).
- Additional storage may need to be allocated for the BOOTPD 191 minidisk, depending on the number of IBM Network Station clients in your environment.

### 5.3.3.2 DASD Storage Requirements for Existing Minidisks

Figure 22. Additional DASD Storage Requirements for Existing Minidisks - IBM Network Station Support

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	2D2	3390 3380 9345	66 79 79	94080	11760	Contains serviced files. (2*) <b>VMSYS:P735FALK.TCPIP.DELTA</b>
P735FALK	491	3390 3380 9345	2 3 3	2880	_____	Test build disk for server code. This code will be copied to a production disk (TCPMAINT 591), so that disk will also require this amount of free space.
TCPMAINT	198	3390 3380 9345	___ ___ ___	_____	_____	Contains configuration files for clients and servers. (3*)
TCPMAINT	591	3390 3380 9345	2 3 3	2880	_____	Production build disk for server code.

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.  
For installation to SFS directories, a total of 11760 4K blocks are required, including the IBM Network Station Client Code (without the IBM Network Station Client Code, 360 4K blocks are required).
2. Additional storage will need to be allocated for the P735FALK 2D2 minidisk if you plan to maintain more than two distinct PTF-level copies of the NSTATION TARBIN file in your environment.
3. For the TCPMAINT 198 minidisk, the changes required for IBM Network Station support are negligible.

### 5.3.4 DASD Storage Requirements for DHCPD Server Support

The storage requirements listed in Figure 23 and Figure 24 on page 31 are applicable only if you intend to install the corrective service for DHCPD server support. See Appendix F, "DHCPD Server Support" on page 197 for more information about these and other DHCPD server requirements.

#### 5.3.4.1 DASD Storage Requirements for Target Minidisks

Figure 23. DASD Storage Requirements for Target Minidisks - DHCPD Server Support

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
DHCPD	191	3390	2	2400	_____	DHCPD user ID 191 minidisk. (2*)
		3380	2			
		9345	2			

**Notes:**

- Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.  
11400 4K blocks are needed for SFS install, including the IBM Network Station Client Code (10 4K blocks without the IBM Network Station Client Code).
- Additional storage may need to be allocated for the DHCPD 191 minidisk, depending on the number of IBM Network Station clients in your environment.

### / 5.3.4.2 DASD Storage Requirements for Existing Minidisks

/ Figure 24. Additional DASD Storage Requirements for Existing Minidisks - DHCPD Server Support

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	2D2	3390 3380 9345	2 2 2	2400	300	Contains serviced files. <b>VMSYS:P735FALK.TCPIP.DELTA</b>
P735FALK	491	3390 3380 9345	2 2 2	2400	_____	Test build disk for server code. This code will be copied to a production disk (TCPMAINT 591), so that disk will also require this amount of free space.
TCPMAINT	198	3390 3380 9345	___ ___ ___	_____	_____	Contains configuration files for clients and servers. (2*)
TCPMAINT	591	3390 3380 9345	2 2 2	2400	_____	Production build disk for server code.

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.  
  
11760 4K blocks are needed for SFS install, including the IBM Network Station Client Code (360 4K blocks without the IBM Network Station Client Code).
2. For the TCPMAINT 198 minidisk, the changes required for DHCPD server support are negligible.

### 5.3.5 DASD Requirements for NFS Feature

Figure 25. DASD Storage Requirements for Target Minidisks - NFS Feature

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	3A6	3390 3380 9345	9 10 10	12000	1500	Contains AUX files and software inventory tables that represent the test service level of NFS Feature. <b>VMSYS:P735FALK.VMNFS.APPLYALT</b>
P735FALK	3A2	3390 3380 9345	9 10 10	12000	1500	Contains AUX files and software inventory tables that represent the production service level of NFS Feature. <b>VMSYS:P735FALK.VMNFS.APPLYPROD</b>
VMNFS	191	3390 3380 9345	9 10 10	12000	_____	VMNFS user ID 191 minidisk.

**Note:** Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.

For installation to SFS directories, a total of 3000 4K blocks are required.

Figure 26. Additional DASD Storage Requirements for Existing Minidisks - NFS Feature

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	2B2	3390 3380 9345	3 3 3	3600	450	Contains all the base code shipped with TCP/IP V2 R4 for VM. <b>VMSYS:P735FALK.TCPIP.OBJECT</b>
P735FALK	2B3	3390 3380 9345	3 3 3	3600	450	Contains source files - in <b>PACKED</b> format <b>VMSYS:P735FALK.TCPIP.SOURCE</b>
P735FALK	2B3	3390 3380 9345	5 5 5	6000	750	Contains source files - in <b>UNPACKED</b> format <b>VMSYS:P735FALK.TCPIP.SOURCE</b>
P735FALK	2D2	3390 3380 9345	5 5 5	6000	750	Contains serviced files. <b>VMSYS:P735FALK.TCPIP.DELTA</b>
P735FALK	491	3390 3380 9345	3 3 3	3600	_____	Test build disk for server code. This code will be copied to a production disk (TCPMAINT 591), so that disk will also require this amount of free space.
P735FALK	492	3390 3380 9345	1 1 1	1200	_____	Test build disk for client code. This code will be copied to a production disk (TCPMAINT 592), so that disk will also require this amount of free space.
TCPMAINT	591	3390 3380 9345	3 3 3	3600	_____	Production build disk for server code.
TCPMAINT	592	3390 3380 9345	1 1 1	1200	_____	Production build disk for client code. (2*)

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.
2. For installation to SFS directories, a total of 1650 4K blocks are required. This total accounts for installation of packed source files. If source files will be unpacked, 1950 4K blocks are required.
3. See Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182 for information about copying client code to the Product Code minidisk.

### 5.3.6 DASD Requirements for Kerberos US Feature

*Figure 27. DASD Storage Requirements for Target Minidisks - Kerberos US Feature*

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	4A6	3390 3380 9345	9 10 10	12000	1500	Contains AUX files and software inventory tables that represent the test service level of Kerberos US Feature. <b>VMSYS:P735FALK.VMKERB.APPLYALT</b>
P735FALK	4A2	3390 3380 9345	9 10 10	12000	1500	Contains AUX files and software inventory tables that represent the production service level of Kerberos US Feature. <b>VMSYS:P735FALK.VMKERB.APPLYPROD</b>
ADMSERV	191	3390 3380 9345	5 5 5	6000	_____	ADMSERV user ID 191 minidisk.
VMKERB	191	3390 3380 9345	6 7 7	8400	_____	VMKERB user ID 191 minidisk.

**Note:** Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.

For installation to SFS directories, a total of 3000 4K blocks are required.



Figure 28. Additional DASD Storage Requirements for Existing Minidisks - Kerberos US Feature

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	2B2	3390 3380 9345	3 3 3	3600	450	Contains all the base code shipped with TCP/IP V2 R4 for VM. <b>VMSYS:P735FALK.TCPIP.OBJECT</b>
P735FALK	2B3	3390 3380 9345	1 1 1	1200	150	Contains source files - in <b>PACKED</b> format <b>VMSYS:P735FALK.TCPIP.SOURCE</b>
P735FALK	2B3	3390 3380 9345	1 1 1	1200	150	Contains source files - in <b>UNPACKED</b> format <b>VMSYS:P735FALK.TCPIP.SOURCE</b>
P735FALK	2D2	3390 3380 9345	5 5 5	6000	750	Contains serviced files. <b>VMSYS:P735FALK.TCPIP.DELTA</b>
P735FALK	491	3390 3380 9345	4 4 4	4800	_____	Test build disk for server code. This code will be copied to a production disk (TCPMAINT 591), so that disk will also require this amount of free space.
P735FALK	492	3390 3380 9345	4 4 4	4800	_____	Test build disk for client code. This code will be copied to a production disk (TCPMAINT 592), so that disk will also require this amount of free space.
TCPMAINT	591	3390 3380 9345	4 4 4	4800	_____	Production build disk for server code.
TCPMAINT	592	3390 3380 9345	4 4 4	4800	_____	Production build disk for client code. (2*)

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.
2. For installation to SFS directories, a total of 1350 4K blocks are required. This total accounts for installation of packed source files. If source files will be unpacked, 1350 4K blocks are required.
3. See Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182 for information about copying client code to the Product Code minidisk.

### 5.3.7 DASD Requirements for Kerberos WT Feature

Figure 29. DASD Storage Requirements for Target Minidisks - Kerberos WT Feature

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	4A6	3390 3380 9345	9 10 10	12000	1500	Contains AUX files and software inventory tables that represent the test service level of Kerberos WT Feature. <b>VMSYS:P735FALK.VMKERB.APPLYALT</b>
P735FALK	4A2	3390 3380 9345	9 10 10	12000	1500	Contains AUX files and software inventory tables that represent the production service level of Kerberos WT Feature. <b>VMSYS:P735FALK.VMKERB.APPLYPROD</b>
ADMSERV	191	3390 3380 9345	5 5 5	6000	_____	ADMSERV user ID 191 minidisk.
VMKERB	191	3390 3380 9345	6 7 7	8400	_____	VMKERB user ID 191 minidisk.
<p><b>Note:</b> Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.</p> <p>For installation to SFS directories, a total of 3000 4K blocks are required.</p>						

Figure 30. Additional DASD Storage Requirements for Existing Minidisks - Kerberos WT Feature

Minidisk owner (User ID)	Default Address	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage
		DASD	CYLS			Default SFS Directory Name
P735FALK	2B2	3390 3380 9345	3 3 3	3600	450	Contains all the base code shipped with TCP/IP V2 R4 for VM. <b>VMSYS:P735FALK.TCPIP.OBJECT</b>
P735FALK	2B3	3390 3380 9345	1 1 1	1200	150	Contains source files - in <b>PACKED</b> format <b>VMSYS:P735FALK.TCPIP.SOURCE</b>
P735FALK	2B3	3390 3380 9345	1 1 1	1200	150	Contains source files - in <b>UNPACKED</b> format <b>VMSYS:P735FALK.TCPIP.SOURCE</b>
P735FALK	2D2	3390 3380 9345	5 5 5	6000	750	Contains serviced files. <b>VMSYS:P735FALK.TCPIP.DELTA</b>
P735FALK	491	3390 3380 9345	4 4 4	4800	_____	Test build disk for server code. This code will be copied to a production disk (TCPMAINT 591), so that disk will also require this amount of free space.
P735FALK	492	3390 3380 9345	4 4 4	4800	_____	Test build disk for client code. This code will be copied to a production disk (TCPMAINT 592), so that disk will also require this amount of free space.
TCPMAINT	591	3390 3380 9345	4 4 4	4800	_____	Production build disk for server code.
TCPMAINT	592	3390 3380 9345	4 4 4	4800	_____	Production build disk for client code. (2*)

**Notes:**

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.
2. For installation to SFS directories, a total of 1350 4K blocks are required. This total accounts for installation of packed source files. If source files will be unpacked, 1350 4K blocks are required.
3. See Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182 for information about copying client code to the Product Code minidisk.

---

## 6.0 Installation Instructions

This section describes the installation methods and the step-by-step procedures to install and activate the TCP/IP V2 R4 for VM base product, Source Feature, NFS Feature, Kerberos US Feature, and Kerberos World Trade Feature.

The procedures that follow are in two-column format. The steps to be performed are in bold large numbers. Commands for these steps are on the left hand side of the page numbered using bold numerals. Additional information for a command may exist to the right of the command. For more information about the two-column format see "Understanding Dialogs with the System" in the *VM/ESA Installation Guide*.

**Each step of the installation instructions must be followed. Do not skip any step unless directed otherwise.**

Throughout these instructions, the use of IBM-supplied default minidisk addresses and user IDs is assumed. If you use different user IDs, minidisk addresses, or SFS directories to install TCP/IP V2 R4 for VM, adapt these instructions as needed for your environment.

### Note!

The sample console output presented throughout these instructions was produced on a VM/ESA Version 2 Release 2.0 system and assumes that you are installing to minidisks using the default PPF and component names. If you're installing TCP/IP V2 R4 for VM on a different VM/ESA system or using a different PPF or component, the results obtained for some commands may differ from those depicted here.

---

## 6.1 TCP/IP for VM Installation Process Overview

The following is a brief description of the main steps to install the TCP/IP V2 R4 for VM base product and its features using VMSES/E.

- Plan Your Installation

Use the VMFINS command to load several files needed to install the base product or features from the install tape and to obtain resource requirements.

- Allocate Resources

The information obtained from the previous step is used to allocate the appropriate minidisks (or SFS directories) and user IDs needed to install the base product or feature.

- Install the Base Product or Feature

Use the VMFINS command to load the base product or feature files from tape to the test BUILD and BASE minidisks/directories. When you install the base product, you can also optionally load the

sample files to the SAMPLE disk and the softcopy publications to a BASE disk. VMFINS is then used to update the VM SYSBLDS file used by VMSES/E for software inventory management.

- Install the Initial RSU

If a Recommended Service Upgrade (RSU) service tape is shipped with the installation tapes, the service should be applied to the system.

- Place the Base Product or Feature Files into Production

The base product or feature files are copied from the test BUILD disks to production BUILD disks.

- Configure the Base Product or Feature

Information is provided to assist in customizing configuration files needed to run the base product or feature.

For a complete description of all VMSES/E installation options refer to:

- *VMSES/E Introduction and Reference* (SC24-5747 or SC24-5444)

---

## 6.2 Installing TCP/IP V2 R4 for VM

### Quick Index for Installation Instructions

Select the installation instructions you should use, based on the TCP/IP feature you will be working with:

- TCP/IP for VM Installation — Begin with 6.2.1, “Plan Your Installation For TCP/IP V2 R4 for VM”
- Source Feature Installation — Begin with 6.3, “Installing TCP/IP V2 R4 for VM - Source Feature” on page 62
- NFS Feature Installation — Begin with 6.4, “Installing TCP/IP V2 R4 for VM - NFS Feature” on page 69
- Kerberos US Feature Installation — Begin with 6.5, “Installing TCP/IP V2 R4 for VM - Kerberos US Feature” on page 84
- Kerberos WT Feature Installation — Begin with 6.6, “Installing TCP/IP V2 R4 for VM - Kerberos WT Feature” on page 100

### 6.2.1 Plan Your Installation For TCP/IP V2 R4 for VM

The VMFINS command will be used to plan the installation. This is a two-step process that will:

- load the first tape file, which contains installation files
- generate a “PLANINFO” file that lists:
  - all user ID and MDISK (minidisk) requirements
  - required products

To obtain planning information for your environment:

**1** Log on as the TCP/IP V2 R4 for VM installation planner.

This user ID can be any ID that has read access to the MAINT 5E5 minidisk and write access to the MAINT 51D minidisk.

**2** Mount the TCP/IP V2 R4 for VM installation tape and attach it to this user ID at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.

**3** Establish read access to the VMSES/E code.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

**4** Establish write access to the Software Inventory disk.

**link maint 51d 51d mr  
access 51d d**

The MAINT 51D minidisk is where the VMSES/E system-level Software Inventory and other dependent files reside.

**Note:** If another user already has the MAINT 51D minidisk linked in write (R/W) mode, you'll obtain only read (R/O) access to this minidisk. If this occurs, you'll need to have that user re-link the 51D disk in read-only (RR) mode, after which you need to re-issue the above LINK and ACCESS commands. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

**5** Load the TCP/IP V2 R4 for VM product control files to the 51D minidisk.

**vmfins install info (nomemo)**

The NOMEMO option will load the memo from the tape but will not issue a prompt to send it to the system printer. Specify the MEMO option if you want to be prompted for printing the memo.

This command will perform the following:

- load the Memo-to-Users
- load various product control files, including the Product Parameter File (PPF) and the PRODPART files
- create the VMFINS PRODLIST file on your A-disk. This file contains a list of products on the installation tape.

```
VMFINS2760I VMFINS processing started
VMFINS1909I VMFINS PRODLIST created on your A-disk
VMFINS2760I VMFINS processing completed successfully
Ready;
```

**6** Obtain resource planning information for TCP/IP V2 R4 for VM.

**Note:** The product will **not** be loaded by the VMFINS command at this time.

**vmfins install ppf 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} (plan nomemo)**

Use **tcpip** if you plan to install TCP/IP V2 R4 for VM to minidisks or **tcpipsfs** to install it to Shared File System directories. Use **tcpipnoc** or **tcpipsfsnoc** if you plan to install TCP/IP V2 R4 for VM to minidisks or to the Shared File System respectively, but do not intend to service any of the functions that require Language Environment runtime library support to be installed.

The PLAN option indicates that VMFINS will perform requisite checking, plan system resources, and provide an opportunity to override the defaults in the product parameter file.

**You can override any of the following:**

- the name of the product parameter file
- the default product user IDs
- minidisk/directory definitions

**Notes:**

- a. If you change the PPF name, a default user ID, or other parameters via a PPF override, you'll need to use your changed values instead of those indicated (when appropriate), throughout the rest of these installation instructions, as well as those provided for servicing TCP/IP V2 R4 for VM. For example, you'll need to specify your PPF override file name instead of 5735FALK for certain VMSES/E commands.
- b. If you're not familiar with creating PPF overrides using VMFINS, you should review the "Using the Make Override Panel" section in Chapter 3 of the *VMSES/E Introduction and Reference* before you continue.
- c. For more information about changing the VMSYS file pool name, refer to Chapter 3 of the *VMSES/E Introduction and Reference*.

```
VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALK component TCPIP passed requisite checking
Do you want to create an override for 5735FALK TCPIP (prodid 5735FALK)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFRMT2760I VMFRMT processing started
VMFRMT2760I VMFRMT processing completed successfully
VMFPLA1909I 5735FALK PLANINFO created on your A-disk
VMFINS2760I VMFINS processing completed successfully
```



**Note!**

**Complete this step only if you received message VMFREQ2806W.**

If you receive the following VMFREQ2806W message (from specifying either TCPIP or TCPIP SFS), you must do some additional processing.

```
VMFREQ2806W The following requisites for product 5735FALK component TCPIP are
              not satisfied
VMFREQ2806W Type          Product  Component      PTF
              -----
VMFREQ2806W Requisite    5688198E
VMFREQ2806W OR           2VMVMA10
```

This message indicates that 5688198E, the IBM Language Environment for MVS and VM Release 5 product, is a requisite product for TCP/IP V2 R4 for VM. You can install TCP/IP V2 R4 for VM without this requisite, but you will have to install the Language Environment runtime library before you can use or service certain TCP/IP for VM functions.

If you are going to install IBM Language Environment for MVS and VM Release 5, you can continue with installing TCP/IP V2 R4 for VM. IBM Language Environment for MVS and VM Release 5 is in VMSES/E format, so the VMSES/E system inventory will be updated automatically when you install this requisite product.

- 7** Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview install**

## 6.2.2 Allocate Resources for Installing TCP/IP V2 R4 for VM

Use the planning information in the 5735FALK PLANINFO file, created in the **PLAN** step, to:

- Create the P735FALK user directory entry for installation to minidisks

**OR**

- Create the P735FALK user directory entry for installation to SFS directories
- Create other required and optional user directory entries.

### 6.2.2.1 PROFILE Directory Entries

There are two PROFILE directory entries, PROFILE CMSUSER and PROFILE GCSUSER, that you need to add to the system directory; these profiles are illustrated below. Each directory entry supplied for a TCP/IP V2 R4 for VM service virtual machine includes one of these profiles. The profiles are as follows:

```
PROFILE CMSUSER
  IPL CMS
  MACHINE XA
  SPOOL 00C 2540 READER *
  SPOOL 00D 2540 PUNCH A
  SPOOL 00E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 190 190 RR
  LINK MAINT 19D 19D RR
  LINK MAINT 19E 19E RR
```

```
PROFILE GCSUSER
  IPL GCS PARM AUTOLOG
  MACHINE XA
  NAMESAVE GCS
  SPOOL 00C 2540 READER *
  SPOOL 00D 2540 PUNCH A
  SPOOL 00E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 190 190 RR
  LINK MAINT 19D 19D RR
  LINK MAINT 19E 19E RR
```

**Note:** The NAMESAVE GCS statement can be removed if the GCS saved segment is not restricted.

### 6.2.2.2 Installing TCP/IP V2 R4 for VM to Minidisks

**1** Obtain the user directory from the 5735FALK PLANINFO file.

**Note:** The user directory entries are located at the end of the resource section at the bottom of the PLANINFO file. These entries contain all of the links and privilege classes necessary for the TCP/IP V2 R4 for VM user IDs. Use the directory entries found in the PLANINFO file as models for input to your system directory.

**2** Modify the directory entries, using Figure 19 on page 23 to obtain the minidisk requirements. When you make these changes, ensure you account for revised minidisk requirements that may be applicable.

- Add the MDISK statements to the directory entries for P735FALK, TCPMAINT, and TCPIP. These user IDs are required.
  - The directory entry for the TCPIP virtual machine includes the statement: SHARE RELATIVE 3000

For most installations, the relative CPU share allocation of 3000 should be suitable. However, you are free to change this value to conform to local guidelines established for defining server and guest virtual machine share settings.

- Determine which optional servers you wish to install.
  - Add the appropriate MDISK statement to the directory entry of each server.
  - If you install multiple RSLAVE $n$  machines, duplicate the RSLAVE1 directory entry for each one.
  - If you install multiple NDBSRV $nn$  machines, duplicate the NDBSRV01 directory entry for each one and add a link for each 191 disk to the P735FALK directory entry.
- For each server that you choose to not install, remove the link to the server's minidisk(s) from the P735FALK directory entry.
- The following minidisks can be removed:
  - P735FALK 2C2 if you are not going to install the optional TCP/IP V2 R4 for VM sample files
  - P735FALK 2B3 if you do not intend to install any of the TCP/IP V2 R4 for VM features
  - P735FALK 2B4 if you are not going to install the optional TCP/IP V2 R4 for VM softcopy publications
- The TCP/IP V2 R4 for VM features (Source, NFS, US Kerberos, and World Trade Kerberos) share many of the disks used by the base product. If you are going to eventually install any of the TCP/IP V2 R4 for VM features, you might want to increase the sizes of the disks you are going to define by the extra amount required by the feature. See the tables in 5.3, "DASD Storage and User ID Requirements" on page 20 to determine the additional storage needed by each feature you intend to install.

**3** If you're installing TCP/IP V2 R4 for VM on VM/ESA Version 1 Release 2.2, the following additional directory entry changes must be made:

- For the FTPSERVE directory entry, delete the following statements:

```
POSIXOPT QUERYDB ALLOW
POSIXINFO UID 0
```

**4** Update the system directory

- Add the PROFILE CMSUSER and PROFILE GCSUSER entries. They must be located between the last DIRECTORY statement and the first USER statement.

- Add the directory entries for P735FALK, TCPMAINT, TCPIP, and the optional servers you have chosen. Change the passwords from xxxxx to valid passwords, in accordance with your security guidelines.
  - The NCS 195 disk and its test build disk, P735FALK 395, contain administrator code. Access to these disks should be restricted.
  - The end users of NCS all need to have Read access to the NCS 191 disk.

Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

All minidisks must be formatted before you install TCP/IP V2 R4 for VM.

### 6.2.2.3 Installing TCP/IP V2 R4 for VM to SFS Directories

- 1** Obtain the user directory from the 5735FALK PLANINFO file.

**Note:** The user directory entries are located at the end of the resource section at the bottom of the PLANINFO file. These entries contain all of the links and privilege classes necessary for the TCP/IP V2 R4 for VM user IDs. Use the directory entries found in the PLANINFO file as models for input to your system directory.

- 2** Modify the directory entries, using Figure 19 on page 23 to obtain the minidisk requirements for the minidisks that do not have Shared File System directories specified.

- Add the MDISK statements to the directory entries for P735FALK, TCPMAINT, and TCPIP. These user IDs are required.
  - The directory entry for the TCPIP virtual machine includes the statement: `SHARE RELATIVE 3000`  
  
For most installations, the relative CPU share allocation of 3000 should be suitable. However, you are free to change this value to conform to local guidelines established for defining server and guest virtual machine share settings.
- Determine which optional servers you wish to install.
  - Add the appropriate MDISK statement to the directory entry of each server.
  - If you install multiple RSLAVE $n$  machines, duplicate the RSLAVE1 directory entry for each one.

- If you install multiple NDBSRVnn machines, duplicate the NDBSRV01 directory entry for each one and add a link for each 191 disk to the P735FALK directory entry.

- For each server that you choose not to install, remove the link to the server's disk(s) from the P735FALK directory entry.

**3** The TCP/IP V2 R4 for VM features (Source, NFS, US Kerberos, and World Trade Kerberos) share many of the minidisks used by the base product. If you are going to eventually install any of the TCP/IP V2 R4 for VM features, you might want to increase the sizes of the minidisks you are going to define by the extra amount required by the feature. See the tables in 5.3, "DASD Storage and User ID Requirements" on page 20 to determine the additional storage needed by each feature you intend to install.

**4** If you intend to use an SFS directory as the work space for the P735FALK user ID, add IPL CMS PARM FILEPOOL VMSYS: after the INCLUDE CMSUSER statement in the P735FALK directory entry. This will cause CMS to automatically access the P735FALK's top directory as file mode A.

**5** If you're installing TCP/IP V2 R4 for VM on VM/ESA Version 1 Release 2.2, the following additional directory entry changes must be made:

- For the FTPSERVE directory entry, delete the following statements:

```
POSIXOPT QUERYDB ALLOW
POSIXINFO UID 0
```

**6** Update the system directory

- Add the PROFILE CMSUSER and PROFILE GCSUSER entries. They must be located between the last DIRECTORY statement and the first USER statement.
- Add the directory entries for P735FALK, TCPMAINT, TCPIP, and the optional servers you have chosen. Change the passwords from xxxxx to valid passwords, in accordance with your security guidelines.
  - The NCS 195 disk and its test build disk, P735FALK 395, contain administrator code. Access to these disks should be restricted.
  - The end users of NCS all need to have Read access to the NCS 191 disk.

Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

All minidisks must be formatted before you install TCP/IP V2 R4 for VM.

**7** An SFS install will also require the following steps:

- a** Determine the number of 4k blocks that are required for SFS directories by adding up the 4K blocks required for each SFS directory you plan to use.

If you intend to use all of the default TCP/IP V2 R4 for VM SFS directories, the 4K block requirements are summarized in Figure 19 on page 23. This information will be used when enrolling the P735FALK to the VMSYS filepool.

The TCP/IP V2 R4 for VM features (Source, NFS, US Kerberos, and World Trade Kerberos) share many of the directories used by the base product. If you are going to eventually install any of the TCP/IP V2 R4 for VM features, you might want to increase the size of the SFS storage you are going to obtain by the extra amount required by the feature. See the tables in 5.3, "DASD Storage and User ID Requirements" on page 20 to determine the additional storage needed by each feature you intend to install.

- b** Enroll user P735FALK in the VMSYS filepool using the ENROLL USER command:

```
ENROLL USER P735FALK VMSYS: (BLOCKS blocks)
```

where *blocks* is the number of 4k blocks that you calculated in the previous step.

**Note:** This must be done from a user ID that is an administrator for VMSYS filepool.

- c** Determine if there are enough blocks available in the filepool to install TCP/IP V2 R4 for VM. This information can be obtained from the QUERY FILEPOOL STATUS command. Near the end of the output from this command is a list of minidisks in the filepool and the number of blocks free. If the number of blocks free is smaller than the total 4k blocks needed to install TCP/IP V2 R4 for VM you will need to add space to the filepool. See *VM/ESA File Pool Planning, Administration and Operation* or *VM/ESA SFS and CRR Planning, Administration and Operation* for information about adding space to a filepool.
- d** Create the necessary subdirectories listed in the 5735FALK PLANINFO file using the CREATE DIRECTORY command. If necessary, see the *VM/ESA CMS Command Reference* for more information about the CREATE DIRECTORY command.

```
set filepool vmsys:
create directory vmsys:p735falk
create directory vmsys:p735falk.tcpip
create directory dirid
```

*dirid* is the name of the SFS directory you're creating, such as:

```
create directory vmsys:p735falk.tcpip.object
create directory vmsys:p735falk.tcpip.delta
:
```

A complete list of default SFS directories is provided in Figure 19 on page 23.

**Note:** The following directories do not have to be created:

- **VMSYS:P735FALK.TCPIP.SAMPLE** if you are not going to install the optional TCP/IP V2 R4 for VM sample files
- **VMSYS:P735FALK.TCPIP.SOURCE** if you do not intend to install any of the TCP/IP V2 R4 for VM features
- **VMSYS:P735FALK.TCPIP.PUBS** if you are not going to install the optional TCP/IP V2 R4 for VM softcopy publications

### 6.2.3 Install TCP/IP V2 R4 for VM

- 1 Log on the installation user ID, **P735FALK**.
- 2 Create a PROFILE EXEC that will contain the ACCESS commands for the MAINT 5E5 and 51D minidisks. The SET LDRTBLS 25 command is needed to successfully build textlibs during service.

```
xedit profile exec a
====> input /**/
====> input 'access 5e5 b'
====> input 'access 51d d'
====> input 'set ldrtbls 25'
====> file
```

- 3 Execute the profile to access the necessary MAINT minidisks.

profile

- 4 Establish write access to the Software Inventory disk, if it is not already linked R/W.

**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user who has it linked R/W link it as R/O. You then can issue the following commands to obtain R/W access to it.

**link maint 51d 51d mr  
access 51d d**

- 5** Have the TCP/IP V2 R4 for VM installation tape mounted and attached to P735FALK at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 6** Install the base TCP/IP V2 R4 for VM, without the sample files or softcopy publications (which will be installed in later steps).

**Notes:**

- If you've already created a PPF override file, you should specify your override file name after the **PPF** keyword for the following VMFINS command.
- You may be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you're unsure how to respond to a prompt, refer to the "Installing Products with VMFINS" and "Install Scenarios" chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.
- As explained above, there are several TCP/IP V2 R4 for VM services that require Language Environment runtime library support. If you do not intend to install the Language Environment runtime library and use these services, then specify either **tcpipnoc** or **tcpipsfsnoc** as the component name.

**vmfins install ppf 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} (nomemo nolink**

Use **tcpip** for installing the TCP/IP V2 R4 for VM base to minidisks, or **tcpipsfs** for installing to Shared File System directories. Use **tcpipnoc** or **tcpipsfsnoc** if you plan to install the TCP/IP V2 R4 for VM base to minidisks or to the Shared File System respectively, but do not intend to service any of the functions that require Language Environment runtime library support to be installed.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.



```

VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALK component TCPIP passed requisite checking
Do you want to create an override for 5735FALK TCPIP (prodid 5735FALK)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINT2760I VMFINST processing started
VMFLDP2706I 5735FALK TCPIP (prodid 5735FALK) will be processed as a PDI
product
VMFSET2760I VMFSETUP processing started
VMFUTL2205I Minidisk|Directory Assignments:
String Mode Stat Vdev Label/Directory
VMFUTL2205I LOCALMOD E R/W 2C4 TCP2C4
VMFUTL2205I APPLY F R/W 2A6 TCP2A6
VMFUTL2205I G R/W 2A2 TCP2A2
VMFUTL2205I DELTA H R/W 2D2 TCP2D2
VMFUTL2205I BUILD1 I R/W 491 TCP491
VMFUTL2205I BUILD3 J R/W 492 TCP492
VMFUTL2205I BUILD7 K R/W 395 TCP395
VMFUTL2205I BASE1 L R/W 2B2 TCP2B2
VMFUTL2205I ----- A R/W 191 TCP191
VMFUTL2205I ----- B R/O 5E5 MNT5E5
VMFUTL2205I ----- D R/W 51D MNT51D
VMFUTL2205I ----- S R/O 190 MNT190
VMFUTL2205I ----- Y/S R/O 19E MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS TAPE 9400
VMFREC1851I (1 of 8) VMFRCAXL processing AXLIST
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (2 of 8) VMFRCPTF processing PARTLST
VMFRCP2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (3 of 8) VMFRCCOM processing DELTA
VMFRCC2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (4 of 8) VMFRCALL processing APPLY
VMFRC2159I Loading part(s) to APPLY 2A6 (F)
VMFRC2159I Loaded nnn part(s) to APPLY 2A6 (F)
VMFREC1851I (5 of 8) VMFRCALL processing SERVER
VMFRC2159I Loading part(s) to BUILD1 491 (I)
VMFRC2159I Loaded nnn part(s) to BUILD1 491 (I)
VMFREC1851I (6 of 8) VMFRCALL processing NCS195
VMFRC2159I Loading part(s) to BUILD7 395 (K)
VMFRC2159I Loaded nnn part(s) to BUILD7 395 (K)
VMFREC1851I (7 of 8) VMFRCALL processing BASE
VMFRC2159I Loading part(s) to BASE1 2B2 (L)
VMFRC2159I Loaded nnn part(s) to BASE1 2B2 (L)
VMFREC1851I (8 of 8) VMFRCALL processing CLIENT
VMFRC2159I Loading part(s) to BUILD3 492 (J)
VMFRC2159I Loaded nnn part(s) to BUILD3 492 (J)
VMFREC2760I VMFREC processing completed successfully
VMFINT2760I VMFINST processing completed successfully
VMFINS2760I VMFINS processing completed successfully

```

**7** Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview install

### 8 (Optional) Install the TCP/IP V2 R4 for VM sample files.

You can choose to install the optional TCP/IP V2 R4 for VM sample files. The sample files are not needed to run TCP/IP V2 R4 for VM.

#### Notes:

- If you've already created a PPF override file, you should specify your override file name after the **PPF** keyword for the following VMFINS command.
- You may be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you're unsure how to respond to a prompt, refer to the "Installing Products with VMFINS" and "Install Scenarios" chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.

## vmfins install ppf 5735falk {tcpipsamp | tcpipsfssamp} (nomemo nolink

Use **tcpipsamp** to install the TCP/IP V2 R4 for VM sample files to a minidisk or **tcpipsfssamp** to install them to a Shared File System directory.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```

VMFINS2760I VMFINS processing started
VMFSIP2502W PRODID 5735FALK%TCPIPSAMP not found in requisite table VM SYSREQT D
VMFREQ2805I Product 5735FALK component TCPIPSAMP passed requisite checking
Do you want to create an override for 5735FALK TCPIPSAMP (prodid 5735FALK)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINT2760I VMFINST processing started
VMFLDP2706I 5735FALK TCPIPSAMP (prodid 5735FALK) will be processed as a PDI
product
VMFSET2760I VMFSETUP processing started
VMFUTL2205I Minidisk|Directory Assignments:
String Mode Stat Vdev Label/Directory
VMFUTL2205I LOCALSAM E R/W 2C2 TCP2C2
VMFUTL2205I ----- A R/W 191 TCP191
VMFUTL2205I ----- B R/O 5E5 MNT5E5
VMFUTL2205I ----- D R/W 51D MNT51D
VMFUTL2205I ----- S R/O 190 MNT190
VMFUTL2205I ----- Y/S R/O 19E MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS TAPE 9400
VMFREC1851I (1 of 1) VMFRCALL processing SAMPLE
VMFRCA2159I Loading part(s) to LOCALSAM 2C2 (E)
VMFRCA2159I Loaded nnn part(s) to LOCALSAM 2C2 (E)
VMFREC2760I VMFREC processing completed successfully
VMFINT2760I VMFINST processing completed successfully
VMFINS2760I VMFINS processing completed successfully

```

- 9 Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview install

- 10 (Optional) Install the TCP/IP V2 R4 for VM softcopy publications.

You can choose to install the optional TCP/IP V2 R4 for VM softcopy publications. These files are not needed to run TCP/IP V2 R4 for VM.

### Notes:

- If you've already created a PPF override file, you should specify your override file name after the **PPF** keyword for the following VMFINS command.
- You may be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you're unsure how to respond to a prompt, refer to the "Installing

Products with VMFINS” and “Install Scenarios” chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.

### **vmfins install ppf 5735falk {tcpippubs | tcpipsfpubs} (nomemo nolink**

Use **tcpippubs** to install the TCP/IP V2 R4 for VM softcopy publications to a minidisk, or **tcpipsfpubs** to install them to a Shared File System directory.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```
VMFINS2760I VMFINS processing started
VMFSIP2502W PROPID 5735FALK%TCIIPPUBS not found in requisite table VM SYSREQT D
VMFREQ2805I Product 5735FALK component TCIIPPUBS passed requisite checking
Do you want to create an override for 5735FALK TCIIPPUBS (prodid 5735FALK)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINT2760I VMFINST processing started
VMFLDP2706I 5735FALK TCIIPPUBS (prodid 5735FALK) will be processed as a PDI
product
VMFSET2760I VMFSETUP processing started
VMFUTL2205I Minidisk|Directory Assignments:
String Mode Stat Vdev Label/Directory
VMFUTL2205I BASE3 E R/W 2B4 TCP2B4
VMFUTL2205I ----- A R/W 191 TCP191
VMFUTL2205I ----- B R/O 5E5 MNT5E5
VMFUTL2205I ----- D R/W 51D MNT51D
VMFUTL2205I ----- S R/O 190 MNT190
VMFUTL2205I ----- Y/S R/O 19E MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS TAPE 9400
VMFREC1851I (1 of 1) VMFRCALL processing PUBS
VMFRCA2159I Loading part(s) to BASE3 2B4 (E)
VMFRCA2159I Loaded nnn part(s) to BASE3 2B4 (E)
VMFREC2760I VMFREC processing completed successfully
VMFINT2760I VMFINST processing completed successfully
VMFINS2760I VMFINS processing completed successfully
```

**11** Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

### **vmfview install**

### 6.2.3.1 Update Build Status Table for TCP/IP V2 R4 for VM

- 1 Update the VM SYSBLDS software inventory file for TCP/IP V2 R4 for VM.

**vmfins build ppf 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfnoc} (serviced nolink**

Use the same component that you used to install the TCP/IP V2 R4 for VM base product.

The SERVICED option will build any parts that were not already built on the installation tape and update the Software Inventory build status table showing that the product 5735FALK has been built.

### 6.2.4 Install the Initial TCP/IP V2 R4 for VM RSU

If an initial RSU tape was shipped with the installation tapes, the RSU service should be installed on the system at this time. Follow the procedure defined in 7.2.2, "Preventive Service for TCP/IP V2 R4 for VM" on page 119. Return here right after you have completed the build step, 7.2.2.5, "Build Serviced Objects" on page 125. The following section, 6.2.5, "Place TCP/IP V2 R4 for VM Into Production," will then place the product and the initial service into production.

### 6.2.5 Place TCP/IP V2 R4 for VM Into Production

**Note:** If you have not determined which TCP/IP V2 R4 for VM services will be used within your environment, you should do so at this time, before you place TCP/IP for VM files into production using the following procedure.

See *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for detailed information about the services provided by TCP/IP V2 R4 for VM.

#### 6.2.5.1 Copy TCP/IP V2 R4 for VM Files Into Production

- 1 Log on the installation user ID, **P735FALK**.
- 2 Establish read access to the VMSES/E code and the Software Inventory minidisks.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

**link maint 51d 51d rr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

- 3 Establish the correct minidisk access order.

**vmfsetup 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} (link**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you are using services that require Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

- 4** Modify the 5735FALK CATALOG file so that the appropriate files will be placed into production for your environment.

The 5735FALK CATALOG file is used by the TCP2PROD command to copy files to TCP/IP for VM minidisks. See Appendix A, “TCP2PROD” on page 172 for information about the TCP2PROD command and TCP/IP for VM catalog files.

**Notes:**

- a. You should modify both the product runtime file section (TCPRUN) and the customizable sample file (TCPCONFIG) section of the 5735FALK CATALOG at the same time. For reference purposes, files that can be processed using the TCPRUN section of this file are listed in Figure 31 on page 59; those processed using the TCPCONFIG section are listed in Figure 32 on page 60.
- b. To ensure you are notified of any service-related changes to the 5735FALK CATALOG file, changes should be made using a VMSES/E-format local modification. See Appendix B, “Modifying TCP/IP for VM CATALOG Files” on page 178 for more information about how to change 5735FALK CATALOG file in this manner.
- c. When making changes, ensure that only files associated with the servers you have installed are processed.

- 5** Establish a write link to any TCP/IP for VM production or server minidisks which are not yet linked in this mode.

If you did not add the appropriate LINK statements to the P735FALK directory when TCP/IP for VM resources were allocated (based on information in the 5735FALK PLANINFO file), you will need to manually link the TCP/IP production and server minidisks necessary for your environment. See Figure 31 on page 59 and Figure 32 on page 60 for the link addresses you should use. If you have a PPF override that has changed any of these addresses, use your values.

**link tcpipid vdev1 vdev2 mr**

|  
| **Note:** If another user already has a required minidisk linked in write mode  
| (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs,  
| you'll need to have that user re-link the minidisk in read-only mode (RR), and  
| then re-issue the above LINK command. Do not continue with these  
| procedures until a R/W link is established for the minidisk in question.

- 6** Copy TCP/IP for VM files into production using the TCP2PROD command. The files copied via the following command are those identified in the TCPRUN section of the 5735FALK CATALOG file.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} 5735falk tcprun**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you are using services that require Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

### 6.2.5.2 Configure TCP/IP V2 R4 for VM

TCP/IP V2 R4 for VM has been installed with sample configuration files which are used by TCP/IP clients and servers. The sample configuration files necessary for the clients, and any TCP/IP servers you intend to use, need to be copied to the appropriate minidisk, renamed if necessary, and customized for your installation. Use the TCP2PROD command, as described below, to copy these files into production.

See the *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for detailed explanations about how to configure the TCP/IP for VM servers.

**Note:** For step 2 below, it's assumed that the TCPCONFIG section of the 5735FALK CATALOG has been suitably modified, as described in step 4 of 6.2.5.1, "Copy TCP/IP V2 R4 for VM Files Into Production" on page 55. If this is not the case, you should make any necessary changes to the TCPCONFIG section of the 5735FALK CATALOG file before you continue with the following steps.

- 1 If necessary, establish the appropriate environment, as described by steps 1 through 3, in 6.2.5.1, “Copy TCP/IP V2 R4 for VM Files Into Production” on page 55.
- 2 Copy TCP/IP for VM configuration files into production using the TCP2PROD command. For reference purposes, files that can be processed using the TCPCONFIG section are listed in Figure 32 on page 60.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} 5735falk tcpconfig**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you are using services that require Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

For information about copying client code to the Product Code minidisk, see Appendix C, “Copying TCP/IP for VM Client Code to the Y-Disk” on page 182.

### **6.2.5.3 TCP/IP V2 R4 for VM Product and Sample Configuration Files**

Figure 31 on page 59 lists the name and location of TCP/IP for VM product runtime files provided by IBM, and the names and location of these files after they've been placed into production. The servers which use a given file have been identified as well.



Figure 31. Files to Copy into Production - TCP/IP V2 R4 for VM

P735FALK Test Disk	P735FALK Link Address	Original File Name/Type	Production File Name/Type	Server Minidisk
491	591	—all files—	no change	———
492	592	—all files—	no change	———
395	26D	—all files—	no change	NCS 195
491	261	TCPIPPRO EXEC	PROFILE EXEC	TCPIP 191
491	262	FTPSEPRO EXEC	PROFILE EXEC	FTPSERVE 191
491	263	SMTPPRO EXEC	PROFILE EXEC	SMTP 191
491	264	NAMESPRO EXEC	PROFILE EXEC	NAMESRV 191
491	265	REXECPRO EXEC	PROFILE EXEC	REXECD 191
491	266	X25IPPRO GCS	PROFILE GCS	X25IPI 191
491	267	PORTMPRO EXEC	PROFILE EXEC	PORTMAP 191
491	268	NDBPMPRO EXEC	PROFILE EXEC	NDBPMGR 191
491	269	NDBSNPRO EXEC (1*)	PROFILE EXEC	NDBSRV01 191
491	26A	SNMPQPRO EXEC	PROFILE EXEC	SNMPQE 191
491	26B	SNMPDPRO EXEC	PROFILE EXEC	SNMPD 191
491	26C	NCSPRO EXEC	PROFILE EXEC	NCS 191
492	26C	NCS SOCK1 H	SOCKET H	NCS 191
492	26C	NCS RPC H	RPC H	NCS 191
492	26C	NCS UTIL C	UTIL C	NCS 191
491	26E	NCSLLPRO EXEC	PROFILE EXEC	NCSLLBD 191
491	26F	NCSGLPRO EXEC	PROFILE EXEC	NCSGLBD 191
491	270	ROUTEPRO EXEC	PROFILE EXEC	ROUTED 191
491	271	LPSEPRO EXEC	PROFILE EXEC	LPSEERVE 191
491	272	SNALNPRO GCS	PROFILE GCS	SNALNKA 191

**Notes:**

1. NDBSNPRO EXEC should also be copied to any additional NDBSRV $nn$  servers you have installed.

Figure 32 on page 60 lists the name and location of the TCP/IP for VM sample files as provided by IBM, and the default names and location of these files after they've been customized. The servers which use a given file have been identified as well.

<i>Figure 32 (Page 1 of 2). Files to Customize - TCP/IP V2 R4 for VM</i>				
<b>Sample Disk</b>	<b>Configured Disk</b>	<b>Sample File Name/Type</b>	<b>Configured File Name/Type</b>	<b>Server</b>
591	198	TCPIPXIT EXEC	no change	TCPIP
591	198	PROFILE STCPIP	PROFILE TCPIP	TCPIP
592	198	HOSTS SLOCAL	HOSTS LOCAL	TCPIP
592	198	SCEXIT SEEXEC	SCEXIT EXEC	TCPIP
592	198	SCEXIT ASSEMBLE	no change	TCPIP
592	592	TCPIP SDATA	TCPIP DATA	TCPIP
591	198	FTPDEXIT EXEC	no change	FTPSSERVE
591	198	CHKIPADR SEEXEC	CHKIPADR EXEC	FTPSSERVE
591	198	FTPDRACF EXEC	no change	FTPSSERVE
591	198	VALIDATE EXEC (1*)	no change	FTPSSERVE
591	198	RACFLINK EXEC	no change	FTPSSERVE
592	592,198	FTP SDATA (2*)	FTP DATA	FTPSSERVE
592	592	FTPPERM EXEC (3*)	no change	FTPSSERVE
591	198	SMTPEXIT EXEC	no change	SMTP
591	198	SMTP SCONFIG	SMTP CONFIG	SMTP
591	198	SECURITY MEMO	no change	SMTP
591	198	SMTP SECTABLE	no change	SMTP
592	592	SMTP SDATA	SMTP DATA	SMTP
592	592	NOTE EXEC TCP (4*)	NOTE EXEC	SMTP
592	592	SENDFILE EXEC TCP (4*)	SENDFILE EXEC	SMTP
591	198	NAMESXIT EXEC	no change	NAMESRV
591	198	NSMAIN SDATA	NSMAIN DATA	NAMESRV
591	198	VALIDUSR SEEXEC	VALIDUSR EXEC	NAMESRV
591	198	REXECXIT EXEC	no change	REXECD
591	198	REXDRACF EXEC	no change	REXECD
591	198	VALIDATE EXEC (1*)	no change	REXECD
591	198	X25IPI SCONFIG	X25IPI CONFIG	X25IPI
<b>Notes:</b>				
1. VALIDATE EXEC is used by FTPSSERVE, REXECD, and LPSSERVE servers.				
2. FTP DATA is used by both the FTP server and the FTP client.				
3. FTPPERM EXEC is used by the FTP client, so must be accessible to FTP end users. If this file needs to be modified, this should be done using a VMSES/E-format local modification.				
4. Review Chapter 12 of <i>TCP/IP V2 R4 for VM: Planning and Customization</i> to determine how these files should be installed for your environment.				

Figure 32 (Page 2 of 2). Files to Customize - TCP/IP V2 R4 for VM

Sample Disk	Configured Disk	Sample File Name/Type	Configured File Name/Type	Server
591	198	PORTMXIT EXEC	no change	PORTMAP
591	198	NDBPMXIT EXEC	no change	NDBPMGR
591	198	NDBSNXIT EXEC	no change	NDBSRV01
591	198	SNMPDXIT EXEC	no change	SNMPD
591	198	SNMPQXIT EXEC	no change	SNMPQE
591	198	MIB_DESC DATA	no change	SNMPQE
591	198	NCSLLXIT EXEC	no change	NCSLLBD
591	198	NCSGLXIT EXEC	no change	NCSGLBD
591	198	ROUTEXIT EXEC	no change	ROUTED
591	198	LPDEXIT EXEC	no change	LPSERVE
591	198	LPD SCONFIG	LPD CONFIG	LPSERVE
591	198	VALIDATE EXEC (1*)	no change	LPSERVE

**Notes:**

1. VALIDATE EXEC is used by FTPSERVE, REXECD, and LPSERVE servers.
2. FTP DATA is used by both the FTP server and the FTP client.
3. FTPPERM EXEC is used by the FTP client, so must be accessible to FTP end users. If this file needs to be modified, this should be done using a VMSES/E-format local modification.
4. Review Chapter 12 of *TCP/IP V2 R4 for VM: Planning and Customization* to determine how these files should be installed for your environment.

**TCP/IP V2 R4 for VM is now installed and built on your system.**

---

## 6.3 Installing TCP/IP V2 R4 for VM - Source Feature

### 6.3.1 Plan Your Installation for the Source Feature

The VMFINS command will be used to plan the installation. This section has 2 main steps that will:

- load the first tape file, containing installation files
- generate a “PLANINFO” file listing
  - all user ID and MDISK (minidisk) requirements
  - required products

To obtain planning information for your environment:

- 1** Log on the installation user ID, **P735FALK**.
- 2** Mount the Source Feature installation tape and attach it to the user ID at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 3** Establish read access to the VMSES/E code if you do not already have it.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

- 4** Establish write access to the Software Inventory disk if you do not already have it.

**link maint 51d 51d mr  
access 51d d**

The MAINT 51D minidisk is where the VMSES/E system-level Software Inventory and other dependent files reside.

**Note:** If another user already has the MAINT 51D minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the 51D in read-only mode (RR), and then re-issue the above LINK and ACCESS commands. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

- 5** Load the Source Feature product control files to the 51D minidisk.

### vmfins install info (nomemo)

The NOMEMO option will load the memo from the tape but will not issue a prompt to send them to the system printer. Specify the MEMO option if you want to be prompted for printing the memo.

This command will perform the following:

- load Memo-to-Users
- load various product control files, including the Product Parameter File (PPF) and the PRODPART files
- create VMFINS PRODLIST on your A-disk. The VMFINS PRODLIST contains a list of products on the installation tape.

```
VMFINS2760I VMFINS processing started
VMFINS1909I VMFINS PRODLIST created on your A-disk
VMFINS2760I VMFINS processing completed successfully
Ready;
```

### 6 Obtain resource planning information for the Source Feature.

**Note:** The product will **not** be loaded by the VMFINS command at this time.

### vmfins install ppf 5735falm {source | sourcesfs} (plan nomemo)

Use **source** if you plan to install the Source Feature to a minidisk, or **sourcesfs** to install it to a Shared File System directory.

The PLAN option indicates that VMFINS will perform requisite checking, plan system resources, and provide an opportunity to override the defaults in the product parameter file.

**You can override any of the following:**

- the name of the product parameter file
- the default user IDs
- minidisk/directory definitions

**Notes:**

- a. If you change the PPF name, a default user ID, or other parameters via a PPF override, you'll need to use your changed values instead of those indicated (when appropriate), throughout the rest of the installation instructions. For example, you'll need to specify your PPF override file name instead of 5735FALM for certain VMSES/E commands.
- b. If you're not familiar with creating PPF overrides using VMFINS, you should review the "Using the Make Override Panel" section in Chapter 3 of the *VMSES/E Introduction and Reference* before you continue.
- c. For more information about changing the VMSYS file pool name, refer to Chapter 3 of the *VMSES/E Introduction and Reference*.

```
VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALM component SOURCE passed requisite checking
Do you want to create an override for 5735FALM SOURCE (prodid 5735FALM)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFRMT2760I VMFRMT processing started
VMFRMT2760I VMFRMT processing completed successfully
VMFPLA1909I 5735FALM PLANINFO created on your A-disk
VMFINS2760I VMFINS processing completed successfully
```

**Note!**

**Complete this step only if you received message VMFREQ2806W.**

If you receive the following VMFREQ2806W message (specifying either SOURCE or SOURCESFS), you must do some additional processing.

```
VMFREQ2806W The following requisites for product 5735FALM component SOURCE are
not satisfied
VMFREQ2806W Type          Product  Component      PTF
-----
VMFREQ2806W Prerequisite  5735FALK
```

This message indicates that you have not yet installed the base product, TCP/IP V2 R4 for VM. You must install the base before you install the Source Feature. Then return here to complete the installation of the Source Feature.

- 7** Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview install**

## 6.3.2 Allocate Resources for Installing the Source Feature

Use the planning information in the 5735FALM PLANINFO file, created in the **PLAN** step, to enlarge the Source disk or directory, if it isn't the appropriate size.

### 6.3.2.1 Installing the Source Feature to Minidisks

- 1 Obtain the disk information from the 5735FALM PLANINFO file. If you intend to unpack the source files after you install them, use the information in Figure 20 on page 27 to determine the necessary size of the minidisk.
- 2 Update the system directory
  - Modify the directory entry for P735FALK to contain a source minidisk of the appropriate size.
  - Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

The Source minidisk must be formatted before installing the Source Feature.

### 6.3.2.2 Installing the Source Feature to SFS Directories

- 1 Obtain the disk information from the 5735FALM PLANINFO file.
- 2 The number of 4k blocks that are required for the source SFS directory in both packed and unpacked format is specified in Figure 20 on page 27. This information will be used when enrolling the P735FALK to the VMSYS filepool.
- 3 Enroll user P735FALK in the VMSYS filepool if not already done. Use the ENROLL USER command:

```
ENROLL USER P735FALK VMSYS: (BLOCKS blocks)
```

where *blocks* is the number of 4k blocks that you calculated in the previous step.

**Note:** This must be done from a user ID that is an administrator for VMSYS: filepool.

- 4 If P735FALK was already enrolled in the VMSYS filepool, you might need to increase P735FALK's storage to hold the source files. Use the MODIFY USER command:

```
MODIFY USER +blocks FOR P735FALK VMSYS:
```

where *blocks* is the number of 4k blocks that you calculated in the previous step.

**Note:** This must be done from a user ID that is an administrator for VMSYS: filepool.

- 5** Determine if there are enough blocks available in the filepool to install the Source Feature. This information can be obtained from the QUERY FILEPOOL STATUS command. Near the end of the output from this command is a list of minidisks in the filepool and the number of blocks free. If the number of blocks free is smaller than the total 4k blocks needed to install the Source Feature you will need to add space to the filepool. See *VM/ESA File Pool Planning, Administration and Operation* or *VM/ESA SFS and CRR Planning, Administration and Operation* for information about adding space to a filepool.
- 6** Create the necessary subdirectory listed in the 5735FALM PLANINFO file using the CREATE DIRECTORY command. If necessary, see the *VM/ESA CMS Command Reference* for more information about the CREATE DIRECTORY command.

```
set filepool vmsys:  
create directory vmsys:p735falk.tcpip.source
```

### 6.3.3 Install the Source Feature

- 1** Log on the installation user ID, **P735FALK**.
- 2** Make sure you have the MAINT 5E5 minidisk accessed as B and the MAINT 51D minidisk (the Software Inventory disk) accessed in write mode as D.  
**Note:** If you cannot link and access the MAINT 51D minidisk R/W, you will need to have the user who has it linked R/W link it as R/O. You then can issue the following commands to obtain R/W access to it.

```
link maint 51d 51d mr  
access 51d d
```

- 3** Have the Source Feature installation tape mounted and attached to P735FALK at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 4** Install the Source Feature.



**Notes:**

- If you've already created a PPF override file, you should specify your override file name after the **PPF** keyword for the following VMFINS command.
- You may be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you're unsure how to respond to a prompt, refer to the “Installing Products with VMFINS” and “Install Scenarios” chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.

**vmfins install ppf 5735falm {source | sourcesfs} (nomemo nolink**

Use **source** to install the Source Feature to a minidisk, or **sourcesfs** to install it to a Shared File System directory.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```
VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALM component SOURCE passed requisite checking
Do you want to create an override for 5735FALM SOURCE (prodid 5735FALM)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINT2760I VMFINST processing started
VMFLDP2706I 5735FALM SOURCE (prodid 5735FALM) will be processed as a PDI
product
VMFSET2760I VMFSETUP processing started
VMFUTL2205I Minidisk|Directory Assignments:
String      Mode  Stat  Vdev  Label/Directory
VMFUTL2205I BASE2   E    R/W  2B3   TCP2B3
VMFUTL2205I ----- A    R/W  191   TCP191
VMFUTL2205I ----- B    R/O  5E5   MNT5E5
VMFUTL2205I ----- D    R/W  51D   MNT51D
VMFUTL2205I ----- S    R/O  190   MNT190
VMFUTL2205I ----- Y/S  R/O  19E   MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS TAPE 9400
VMFREC1851I (1 of 1) VMFRCALL processing SOURCE
VMFRCA2159I Loading part(s) to BASE2 2B3 (E)
VMFRCA2159I Loaded nnn part(s) to BASE2 2B3 (E)
VMFREC2760I VMFREC processing completed successfully
VMFINT2760I VMFINST processing completed successfully
VMFINS2760I VMFINS processing completed successfully
```

- 5 Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview install**

## 6.3.4 Place the Source Feature Into Production

The Source Feature does not contain any executable files, so there is nothing to put into production or configure. The source files remain on the Source disk.

### 6.3.4.1 Unpack the Source Feature

The Source Feature has been installed with packed source files. If you wish to unpack the source files, log on P735FALK and invoke the copy command with the unpack option. The Source disk must have been defined large enough to store the unpacked files.

**access 2b3 e**  
**copyfile fn ft e (unpack olddate replace**

Access the Source directory if you installed to the Shared File System.

If **all** files on the disk are packed, you can unpack them all by specifying \* \* for *fn ft*; otherwise, you should unpack Source Feature files on an individual basis.

**The Source Feature is now installed and built on your system.**

---

## 6.4 Installing TCP/IP V2 R4 for VM - NFS Feature

### 6.4.1 Plan Your Installation for the NFS Feature

The VMFINS command will be used to plan the installation. This section has 2 main steps that will:

- load the first tape file, containing installation files
- generate a “PLANINFO” file listing
  - all user ID and MDISK (minidisk) requirements
  - required products

To obtain planning information for your environment:

- 1** Log on the installation user ID, **P735FALK**.
- 2** Mount the NFS Feature installation tape and attach it to the user ID at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 3** Establish read access to the VMSES/E code.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

- 4** Establish write access to the Software Inventory disk.

**link maint 51d 51d mr  
access 51d d**

The MAINT 51D minidisk is where the VMSES/E system-level Software Inventory and other dependent files reside.

**Note:** If another user already has the MAINT 51D minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the 51D in read-only mode (RR), and then re-issue the above LINK and ACCESS commands. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

- 5** Load the NFS Feature product control files to the 51D minidisk.

## vmfins install info (nomemo)

The NOMEMO option will load the memo from the tape but will not issue a prompt to send them to the system printer. Specify the MEMO option if you want to be prompted for printing the memo.

This command will perform the following:

- load Memo-to-Users
- load various product control files, including the Product Parameter File (PPF) and the PRODPART files
- create VMFINS PRODLIST on your A-disk. The VMFINS PRODLIST contains a list of products on the installation tape.

```
VMFINS2760I VMFINS processing started
VMFINS1909I VMFINS PRODLIST created on your A-disk
VMFINS2760I VMFINS processing completed successfully
Ready;
```

## 6 Obtain resource planning information for the NFS Feature.

**Note:** The product will **not** be loaded by the VMFINS command at this time.

## vmfins install ppf 5735fall {vmnfs | vmnfssfs} (plan nomemo)

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product to minidisks or **vmnfssfs** if you installed it to Shared File System directories.

The PLAN option indicates that VMFINS will perform requisite checking, plan system resources, and provide an opportunity to override the defaults in the product parameter file.

**You can override any of the following:**

- the name of the product parameter file
- the default user IDs
- minidisk/directory definitions

**Notes:**

- a. If you change the PPF name, a default user ID, or other parameters via a PPF override, you'll need to use your changed values instead of those indicated (when appropriate), throughout the rest of the installation instructions, as well as those provided for servicing NFS Feature. For example, you'll need to specify your PPF override file name instead of 5735FALL for certain VMSES/E commands.
- b. If you're not familiar with creating PPF overrides using VMFINS, you should review the "Using the Make Override Panel" section in Chapter 3 of the *VMSES/E Introduction and Reference* before you continue.
- c. For more information about changing the VMSYS file pool name, refer to Chapter 3 of the *VMSES/E Introduction and Reference*.

```
VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALL component VMNFS passed requisite checking
Do you want to create an override for 5735FALL VMNFS (prodid 5735FALL)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFRMT2760I VMFRMT processing started
VMFRMT2760I VMFRMT processing completed successfully
VMFPLA1909I 5735FALL PLANINFO created on your A-disk
VMFINS2760I VMFINS processing completed successfully
```

**Note!**

**Complete this step only if you received message VMFREQ2806W.**

If you receive the following VMFREQ2806W message (specifying either VMNFS or VMNFSDFS), you must do some additional processing.

```
VMFREQ2806W The following requisites for product 5735FALL component VMNFS are
              not satisfied
```

```
VMFREQ2806W Type          Product  Component      PTF
              -----
```

```
VMFREQ2806W Prerequisite  5735FALK
```

```
VMFREQ2806W Requisite    5688198E
```

```
VMFREQ2806W OR           2VMVMA10
```

The messages indicate that 5735FALK, product TCP/IP V2 R4 for VM, is a prerequisite product for the NFS Feature. You must install it before you can continue with this installation.

In addition, the messages indicate that 5688198E, the IBM Language Environment for MVS and VM Release 5 product, is a requisite. You can install the NFS Feature without this requisite, but will have to install IBM Language Environment for MVS and VM Release 5 before you run or service the NFS Feature.

If you are going to install IBM Language Environment for MVS and VM Release 5, you can continue with installing NFS Feature. IBM Language Environment for MVS and VM Release 5 is in VMSES/E format, so the VMSES/E system inventory will be updated automatically when you install this requisite product.

- 7** Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview install

### 6.4.2 Allocate Resources for Installing the NFS Feature

Use the planning information in the 5735FALL PLANINFO file, created in the **PLAN** step, to:

- Update the P735FALK user directory entry for minidisk install

**OR**

Update the P735FALK user directory entry for SFS install

- Create the required user directory entry.

### 6.4.2.1 PROFILE Directory Entries

There is a PROFILE directory entry, PROFILE CMSUSER, that is used in the VMNFS directory entry. This profile should already have been added to the system directory when you installed the TCP/IP V2 R4 for VM base product. The profile is as follows:

```
PROFILE CMSUSER
  IPL CMS
  MACHINE XA
  SPOOL 00C 2540 READER *
  SPOOL 00D 2540 PUNCH A
  SPOOL 00E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 190 190 RR
  LINK MAINT 19D 19D RR
  LINK MAINT 19E 19E RR
```

### 6.4.2.2 Installing the NFS Feature to Minidisks

- 1 Obtain the user directory from the 5735FALL PLANINFO file.

**Note:** The user directory entry is located at the end of the resource section at the bottom of the PLANINFO file. This entry contains all of the links and privilege classes necessary for the VMNFS user ID. Use the directory entry found in the PLANINFO file as a model for input to your system directory.

- 2 Modify directory entries, using Figure 25 on page 32 and Figure 26 on page 33 to obtain the minidisk requirements.

- Add the MDISK statements to the directory entries for P735FALK and VMNFS. These user IDs are required. Also add an MDISK statement to P735FALK for the Source disk if it does not already exist. Use one of the two Source disk sizes specified in Figure 26 on page 33 depending on whether you intend to unpack the source files after installation.
- Add the following link to the P735FALK directory entry

```
LINK VMNFS 191 274 MR
```

- Increase the size of the existing P735FALK and TCPMAINT service disks by the amount specified in Figure 26 on page 33 if this has not already been done. Use one of the two Source disk sizes, depending on whether you intend to unpack the source files after installation.

- 3 Update the system directory

- Add the directory entry for VMNFS. Change the password from xxxxx to a valid password, in accordance with your security guidelines.
- Modify the directory entries for P735FALK and TCPMAINT.

Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

All new minidisks must be formatted before you install the NFS Feature.

### 6.4.2.3 Installing the NFS Feature to SFS Directories

- 1 Obtain the user directory from the 5735FALL PLANINFO file.

**Note:** The user directory entry is located at the end of the resource section at the bottom of the PLANINFO file. This entry contains all of the links and privilege classes necessary for the VMNFS user ID. Use the directory entry found in the PLANINFO file as a model for input to your system directory.

- 2 Modify directory entries, using Figure 25 on page 32 and Figure 26 on page 33 to obtain the minidisk requirements for the minidisks that do not have Shared File System directories specified.

- Add the MDISK statements to the directory entry for VMNFS. This user ID is required.

- Add the following link to the P735FALK directory entry

```
LINK VMNFS 191 274 MR
```

- Increase the size of the existing P735FALK and TCPMAINT service disks that are not on the Shared File System by the amount specified in Figure 26 on page 33 if this has not already been done.

- 3 Update the system directory

- Add the directory entry for VMNFS. Change the password from xxxxx to a valid password, in accordance with your security guidelines.
- Modify the directory entries for P735FALK and TCPMAINT.

Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

All new minidisks must be formatted before you install the NFS Feature.

- 4 An SFS install will also require the following steps:

- a Determine the number of 4k blocks that are required for SFS directories by adding up the 4K blocks required for each SFS directory you plan to use. Use one of the two Source directory sizes, depending on whether you intend to unpack the source files after installation.

If you intend to use all of the default NFS Feature SFS directories, the 4K block requirements are summarized in Figure 25 on page 32 and



Figure 26 on page 33. This information will be used to modify the storage allocation for P735FALK in the VMSYS filepool.

- b** Increase the amount of SFS storage allocated to user P735FALK in the VMSYS filepool, if sufficient storage was not already allocated. Use the MODIFY USER command:

```
MODIFY USER +blocks FOR P735FALK VMSYS:
```

where *blocks* is the number of 4k blocks that you calculated in the previous step.

**Note:** This must be done from a user ID that is an administrator for VMSYS: filepool.

- c** Determine if there are enough blocks available in the filepool to install the NFS Feature. This information can be obtained from the QUERY FILEPOOL STATUS command. Near the end of the output from this command is a list of minidisks in the filepool and the number of blocks free. If the number of blocks free is smaller than the total 4k blocks needed to install the NFS Feature you will need to add space to the filepool. See *VM/ESA File Pool Planning, Administration and Operation* or *VM/ESA SFS and CRR Planning, Administration and Operation* for information about adding space to a filepool.
- d** Create the necessary subdirectories listed in the 5735FALL PLANINFO file using the CREATE DIRECTORY command. If necessary, see the *VM/ESA CMS Command Reference* for more information about the CREATE DIRECTORY command.

```
set filepool vmsys:  
create directory vmsys:p735falk.vmnfs  
create directory vmsys:p735falk.vmnfs.applyalt  
create directory vmsys:p735falk.vmnfs.applyprod  
create directory vmsys:p735falk.tcpip.source
```

Only create the Source directory if it does not already exist.

### 6.4.3 Install the NFS Feature

- 1** Log on the installation user ID, **P735FALK**.
- 2** Establish read access to the 5E5 minidisk and write access to the 51D minidisk (Software Inventory), if they are not already linked correctly.

**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user who has it linked R/W link it as R/O. You then can issue the following commands to obtain R/W access to it.

**link maint 51d 51d mr  
access 51d d**

**3** Have the NFS Feature installation tape mounted and attached to P735FALK at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.

**4** Install the NFS Feature.

**Notes:**

- If you've already created a PPF override file, you should specify your override file name after the **PPF** keyword for the following VMFINS command.
- You may be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you're unsure how to respond to a prompt, refer to the "Installing Products with VMFINS" and "Install Scenarios" chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.

**vmfins install ppf 5735fall {vmnfs | vmnfssfs} (nomemo nolink**

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product to minidisks; use **vmnfssfs** if you installed it to Shared File System directories.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```

VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALL component VMNFS passed requisite checking
Do you want to create an override for 5735FALL VMNFS (prodid 5735FALL)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINT2760I VMFINST processing started
VMFLDP2706I 5735FALL VMNFS (prodid 5735FALL) will be processed as a PDI
product
VMFSET2760I VMFSETUP processing started
VMFUTL2205I Minidisk|Directory Assignments:
String Mode Stat Vdev Label/Directory
VMFUTL2205I LOCALMOD E R/W 2C4 TCP2C4
VMFUTL2205I APPLY F R/W 3A6 NFS3A6
VMFUTL2205I G R/W 3A2 NFS3A2
VMFUTL2205I DELTA H R/W 2D2 TCP2D2
VMFUTL2205I BUILD1 I R/W 491 TCP491
VMFUTL2205I BUILD3 J R/W 492 TCP492
VMFUTL2205I BASE1 K R/W 2B2 TCP2B2
VMFUTL2205I BASE2 L R/W 2B3 TCP2B3
VMFUTL2205I ----- A R/W 191 TCP191
VMFUTL2205I ----- B R/O 5E5 MNT5E5
VMFUTL2205I ----- D R/W 51D MNT51D
VMFUTL2205I ----- S R/O 190 MNT190
VMFUTL2205I ----- Y/S R/O 19E MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS TAPE 9430
VMFREC1851I (1 of 8) VMFRCAXL processing AXLIST
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (2 of 8) VMFRCPTF processing PARTLST
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (3 of 8) VMFRCOM processing DELTA
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (4 of 8) VMFRCALL processing APPLY
VMFRCX2159I Loading part(s) to APPLY 3A6 (F)
VMFRCX2159I Loaded nnn part(s) to APPLY 3A6 (F)
VMFREC1851I (5 of 8) VMFRCALL processing NFS
VMFRCX2159I Loading part(s) to BASE1 2B2 (K)
VMFRCX2159I Loaded nnn part(s) to BASE1 2B2 (K)
VMFREC1851I (6 of 8) VMFRCALL processing SERVER
VMFRCX2159I Loading part(s) to BUILD1 491 (I)
VMFRCX2159I Loaded nnn part(s) to BUILD1 491 (I)
VMFREC1851I (7 of 8) VMFRCALL processing CLIENT
VMFRCX2159I Loading part(s) to BUILD3 492 (J)
VMFRCX2159I Loaded nnn part(s) to BUILD3 492 (J)
VMFREC1851I (8 of 8) VMFRCALL processing SOURCE
VMFRCX2159I Loading part(s) to BASE2 2B3 (L)
VMFRCX2159I Loaded nnn part(s) to BASE2 2B3 (L)
VMFREC2760I VMFREC processing completed successfully
VMFINT2760I VMFINST processing completed successfully
VMFINS2760I VMFINS processing completed successfully

```

- 5 Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview install**

### 6.4.3.1 Update Build Status Table for the NFS Feature

- 1 Update the VM SYSBLDS software inventory file for the NFS Feature.

**vmfins build ppf 5735fall {vmnfs | vmnfssfs} (serviced nolink**

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product to minidisks; use **vmnfssfs** if you installed it to Shared File System directories.

The SERVICED option will build any parts that were not built on the installation tape and update the Software Inventory build status table showing that the product 5735FALL has been built.

## 6.4.4 Place the NFS Feature Into Production

### 6.4.4.1 Copy NFS Feature Files Into Production

- 1 Log on the installation user ID, **P735FALK**.
- 2 Establish read access to the VMSES/E code and the Software Inventory minidisks.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

**link maint 51d 51d rr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

- 3 Establish the correct minidisk access order.

**vmfsetup 5735fall {vmnfs | vmnfssfs} (link**

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product to minidisks; use **vmnfssfs** if you installed it to Shared File System directories.

- 4 If necessary, modify the 5735FALL CATALOG file so that the appropriate files will be placed into production for your environment.

The 5735FALL CATALOG file is used by the TCP2PROD command to copy files to TCP/IP for VM minidisks. See Appendix A, "TCP2PROD" on page 172 for information about the TCP2PROD command and TCP/IP for VM catalog files.

**Notes:**

- a. You should modify both the product runtime file section (NFSRUN) and the customizable sample file (NFSCONFIG) section of the 5735FALL CATALOG at the same time. For reference purposes, files that can be processed using the NFSRUN section of this file are listed in Figure 33 on page 82; those processed using the NFSCONFIG section are listed in Figure 34 on page 82.
- b. To ensure you are notified of any service-related changes to the 5735FALL CATALOG file, changes should be made using a VMSES/E-format local modification. See Appendix B, "Modifying TCP/IP for VM CATALOG Files" on page 178 for more information about how to change 5735FALL CATALOG file in this manner.

**5** Establish a write link to any TCP/IP for VM production or server minidisks which are not yet linked in this mode.

If you did not add the appropriate LINK statements to the P735FALK directory when TCP/IP for VM resources were allocated (based on information in the 5735FALL PLANINFO file), you will need to manually link the TCP/IP production and server minidisks necessary for your environment. See Figure 33 on page 82 and Figure 34 on page 82 for the link addresses you should use. If you have a PPF override that has changed any of these addresses, use your values.

```
link tcpipid vdev1 vdev2 mr
```

**Note:** If another user already has a required minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the minidisk in read-only mode (RR), and then re-issue the above LINK command. Do not continue with these procedures until a R/W link is established for the minidisk in question.

**6** Copy NFS Feature files into production using the TCP2PROD command. The files copied via the following command are those identified in the NFSRUN section of the 5735FALL CATALOG file.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735fall {vmnfs | vmnfssfs} 5735fall nfsrun**

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product to minidisks; use **vmnfssfs** if you installed it to Shared File System directories.

#### 6.4.4.2 Configure the NFS Feature

The NFS Feature has been installed with sample configuration files which are used by the VMNFS server. These files need to be copied to the appropriate minidisk, renamed if necessary, and customized for your installation. Use the TCP2PROD command, as described below, to copy these files into production.

See the *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for detailed explanations about how to configure the VMNFS server.

**Note:** For step 2 below, it's assumed that the NFSCONFIG section of the 5735FALL CATALOG has been suitably modified, as described in step 4 of 6.4.4.1, "Copy NFS Feature Files Into Production" on page 78. If this is not the case, you should make any necessary changes to the NFSCONFIG section of the 5735FALL CATALOG file before you continue with the following steps.

- 1** If necessary, establish the appropriate environment, as described by steps 1 through 3, in 6.4.4.1, "Copy NFS Feature Files Into Production" on page 78.
- 2** Copy NFS Feature configuration files into production using the TCP2PROD command. For reference purposes, files that can be processed using the NFSCONFIG section are listed in Figure 34 on page 82.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735fall {vmnfs | vmnfssfs} 5735fall nfsconfig**

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product to minidisks; use **vmnfssfs** if you installed it to Shared File System directories.

For information about copying client code to the Product Code minidisk, see Appendix C, “Copying TCP/IP for VM Client Code to the Y-Disk” on page 182.

### 6.4.4.3 NFS Feature Runtime and Sample Configuration Files

Figure 33 lists the name and location of the NFS Feature runtime files provided by IBM, and the names and location of these files after they've been placed into production.

*Figure 33. Files to Copy into Production - NFS Feature*

<b>P735FALK Test Disk</b>	<b>P735FALK Link Address</b>	<b>Original File Name/Type</b>	<b>Production File Name/Type</b>	<b>Server Minidisk</b>
491	591	IPSASM TEXT	no change	_____
491	591	NFSMASK EXEC	no change	_____
491	591	PRINTLOG MODULE	no change	_____
491	591	TRACEVD CALLID	no change	_____
491	591	TVPRINT MODULE	no change	_____
491	591	VMNFS MODULE	no change	_____
491	591	VMNFSCMS EXEC	no change	_____
491	591	VMNFSMON EXECTCP	no change	_____
491	591	VMNFSPRO EXEC	no change	_____
491	591	VMNFSXIT EXEC	no change	_____
491	274	VMNFSPRO EXEC	PROFILE EXEC	VMNFS 191
2B2	274	VMNFS HISTORY	no change	VMNFS 191
492	592	PS2@BIN MOUNTPW	no change	_____
492	592	RT@BIN MOUNTPW	no change	_____
492	592	370@BIN MOUNTPW	no change	_____
492	592	6000@BIN MOUNTPW	no change	_____
492	592	OS2@BIN MOUNTPW	no change	_____

Figure 34 lists the name and location of the NFS Feature sample files as provided by IBM, and the default names and location of these files after they've been customized.

*Figure 34. Files to Customize - NFS Feature*

<b>Sample Disk</b>	<b>Configured Disk</b>	<b>Sample File Name/Type</b>	<b>Configured File Name/Type</b>
591	198	VMNFSCMS EXEC	no change
591	198	VMNFSMON EXECTCP	VMNFSMON EXEC
591	198	VMNFSXIT EXEC	no change



#### 6.4.4.4 Unpack Source Files

The NFS Feature has been installed with packed source files. If you wish to unpack the source files, log on P735FALK and invoke the copy command with the unpack option. The Source disk must have been defined large enough to store the unpacked files.

**access 2b3 m**  
**copyfile *fn ft* m (unpack olddate replace**

Access the Source directory if you installed to the Shared File System.

If **all** files on the disk are packed, you can unpack them all by specifying \* \* for *fn ft*; otherwise, you should unpack NFS Feature files on an individual basis.

**The NFS Feature is now installed and built on your system.**

---

## 6.5 Installing TCP/IP V2 R4 for VM - Kerberos US Feature

### 6.5.1 Plan Your Installation for the Kerberos US Feature

The VMFINS command will be used to plan the installation. This section has 2 main steps that will:

- load the first tape file, containing installation files
- generate a “PLANINFO” file listing
  - all user ID and MDISK (minidisk) requirements
  - required products

To obtain planning information for your environment:

- 1** Log on the installation user ID, **P735FALK**.
- 2** Mount the Kerberos US Feature installation tape and attach it to P735FALK ID at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 3** Establish read access to the VMSES/E code.

**link maint 5e5 5e5 rr**  
**access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

- 4** Establish write access to the Software Inventory disk.

**link maint 51d 51d mr**  
**access 51d d**

The MAINT 51D minidisk is where the VMSES/E system-level Software Inventory and other dependent files reside.

**Note:** If another user already has the MAINT 51D minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the 51D in read-only mode (RR), and then re-issue the above LINK and ACCESS commands. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

- 5** Load the Kerberos US Feature product control files to the 51D minidisk.

### vmfins install info (nomemo)

The NOMEMO option will load the memo from the tape but will not issue a prompt to send them to the system printer. Specify the MEMO option if you want to be prompted for printing the memo.

This command will perform the following:

- load Memo-to-Users
- load various product control files, including the Product Parameter File (PPF) and the PRODPART files
- create VMFINS PRODLIST on your A-disk. The VMFINS PRODLIST contains a list of products on the installation tape.

```
VMFINS2760I VMFINS processing started
VMFINS1909I VMFINS PRODLIST created on your A-disk
VMFINS2760I VMFINS processing completed successfully
Ready;
```

### 6 Obtain resource planning information for the Kerberos US Feature.

**Note:** The product will **not** be loaded by the VMFINS command at this time.

### vmfins install ppf 5735fain {vmkerb | vmkerbsfs} (plan nomemo)

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product to minidisks or **vmkerbsfs** if you installed it to Shared File System directories.

The PLAN option indicates that VMFINS will perform requisite checking, plan system resources, and provide an opportunity to override the defaults in the product parameter file.

**You can override any of the following:**

- the name of the product parameter file
- the default user IDs
- minidisk/directory definitions

**Notes:**

- a. If you change the PPF name, a default user ID, or other parameters via a PPF override, you'll need to use your changed values instead of those indicated (when appropriate), throughout the rest of the installation instructions, as well as those provided for servicing Kerberos US Feature. For example, you'll need to specify your PPF override file name instead of 5735FALN for certain VMSES/E commands.
- b. If you're not familiar with creating PPF overrides using VMFINS, you should review the "Using the Make Override Panel" section in Chapter 3 of the *VMSES/E Introduction and Reference* before you continue.
- c. For more information about changing the VMSYS file pool name, refer to Chapter 3 of the *VMSES/E Introduction and Reference*.

```
VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALN component VMKERB passed requisite checking
Do you want to create an override for 5735FALN VMKERB (prodid 5735FALN)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFRMT2760I VMFRMT processing started
VMFRMT2760I VMFRMT processing completed successfully
VMFPLA1909I 5735FALN PLANINFO created on your A-disk
VMFINS2760I VMFINS processing completed successfully
```

**Note!**

**Complete this step only if you received message VMFREQ2806W.**

If you receive the following VMFREQ2806W message (specifying either VMKERB or VMKERBSFS), you must do some additional processing.

```
VMFREQ2806W The following requisites for product 5735FALN component VMKERB are
              not satisfied
VMFREQ2806W Type          Product  Component      PTF
              -----
VMFREQ2806W Prerequisite  5735FALK
VMFREQ2806W Requisite    5688198E
VMFREQ2806W OR           2VMVMA10
```

The messages indicate that 5735FALK, product TCP/IP V2 R4 for VM, is a prerequisite product for the Kerberos US Feature. You must install it before you can continue with this installation.

In addition, the messages indicate that 5688198E, the IBM Language Environment for MVS and VM Release 5 product, is a requisite. You can install the Kerberos US Feature without this requisite, but will have to install IBM Language Environment for MVS and VM Release 5 before you run or service the Kerberos US Feature.

If you are going to install IBM Language Environment for MVS and VM Release 5, you can continue with installing Kerberos US Feature. IBM Language Environment for MVS and VM Release 5 is in VMSES/E format, so the VMSES/E system inventory will be updated automatically when you install this requisite product.

- 7** Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview install

### 6.5.2 Allocate Resources for Installing the Kerberos US Feature

Use the planning information in the 5735FALN PLANINFO file, created in the **PLAN** step, to:

- Update the P735FALK user directory entry for minidisk install

**OR**

Update the P735FALK user directory entry for SFS install

- Create the other required user directory entries.

### 6.5.2.1 PROFILE Directory Entries

There is a PROFILE directory entry, PROFILE CMSUSER, that is used in the new directory entries. This profile should already have been added to the system directory when you installed the TCP/IP V2 R4 for VM base product. The profile is as follows:

```
PROFILE CMSUSER
  IPL CMS
  MACHINE XA
  SPOOL 00C 2540 READER *
  SPOOL 00D 2540 PUNCH A
  SPOOL 00E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 190 190 RR
  LINK MAINT 19D 19D RR
  LINK MAINT 19E 19E RR
```

### 6.5.2.2 Installing the Kerberos US Feature to Minidisks

**1** Obtain the user directories from the 5735FALN PLANINFO file.

**Note:** The user directory entries are located at the end of the resource section at the bottom of the PLANINFO file. These entries contain all of the links and privilege classes necessary for the Kerberos US Feature user IDs. Use the directory entries found in the PLANINFO file as models for input to your system directory.

**2** Modify the directory entries, using Figure 27 on page 34 and Figure 28 on page 35 to obtain the minidisk requirements.

- Add the MDISK statements to the directory entries for P735FALK, VMKERB, and ADMSERV. These user IDs are required. Also add an MDISK statement to P735FALK for the Source disk if it does not already exist. Use one of the two Source disk sizes specified in Figure 28 on page 35 depending on whether you intend to unpack the source files after installation.

- Add the following links to the P735FALK directory entry

```
LINK VMKERB 191 276 MR
LINK ADMSERV 191 277 MR
```

- Increase the size of the existing P735FALK and TCPMAINT service disks by the amount specified in Figure 28 on page 35 if this has not already been done. Use one of the two Source disk sizes, depending on whether you intend to unpack the source files after installation.

**3** Update the system directory

- Add the directory entries for VMKERB and ADMSERV. Change the passwords from xxxxx to a valid password, in accordance with your security guidelines.
- Modify the directory entries for P735FALK and TCPMAINT.

Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

All new minidisks must be formatted before you install the Kerberos US Feature.

### 6.5.2.3 Installing the Kerberos US Feature to SFS Directories

**1** Obtain the user directory from the 5735FALN PLANINFO file.

**Note:** The user directory entries are located at the end of the resource section at the bottom of the PLANINFO file. These entries contain all of the links and privilege classes necessary for the Kerberos US Feature user IDs. Use the directory entries found in the PLANINFO file as models for input to your system directory.

**2** Modify the directory entries, using Figure 27 on page 34 and Figure 28 on page 35 to obtain the minidisk requirements for the minidisks that do not have Shared File System directories specified.

- Add the MDISK statements to the directory entries VMKERB and ADMSERV. These user IDs are required.
- Add the following links to the P735FALK directory entry

```
LINK VMKERB 191 276 MR
LINK ADMSERV 191 277 MR
```

- Increase the size of the existing P735FALK and TCPMAINT service disks that are not on the Shared File System by the amount specified in Figure 28 on page 35 if this has not already been done.

**3** Update the system directory

- Add the directory entries for VMKERB and ADMSERV. Change the passwords from xxxxx to a valid password, in accordance with your security guidelines.
- Modify the directory entries for P735FALK and TCPMAINT.

Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

All new minidisks must be formatted before you install the Kerberos US Feature.

**4** An SFS install will also require the following steps:

- a** Determine the number of 4k blocks that are required for SFS directories by adding up the 4K blocks required for each SFS directory you plan to use. Use one of the two Source disk sizes, depending on whether you intend to unpack the source files after installation.

If you intend to use all of the default Kerberos US Feature SFS directories, the 4K block requirements are summarized in Figure 27 on page 34 and Figure 28 on page 35. This information will be used to modify the storage allocation for P735FALK in the VMSYS filepool.

- b** Increase the amount of SFS storage allocated to user P735FALK in the VMSYS filepool, if sufficient storage was not already allocated. Use the MODIFY USER command:

```
MODIFY USER +blocks FOR P735FALK VMSYS:
```

where *blocks* is the number of 4k blocks that you calculated in the previous step.

**Note:** This must be done from a user ID that is an administrator for VMSYS: filepool.

- c** Determine if there are enough blocks available in the filepool to install the Kerberos US Feature. This information can be obtained from the QUERY FILEPOOL STATUS command. Near the end of the output from this command is a list of minidisks in the filepool and the number of blocks free. If the number of blocks free is smaller than the total 4k blocks needed to install the Kerberos US Feature you will need to add space to the filepool. See *VM/ESA File Pool Planning, Administration and Operation* or *VM/ESA SFS and CRR Planning, Administration and Operation* for information about adding space to a filepool.

- d** Create the necessary subdirectories listed in the 5735FALN PLANINFO file using the CREATE DIRECTORY command. If necessary, see the *VM/ESA CMS Command Reference* for more information about the CREATE DIRECTORY command.



```
set filepool vmsys:  
create directory vmsys:p735falk.vmkerb  
create directory vmsys:p735falk.vmkerb.applyalt  
create directory vmsys:p735falk.vmkerb.applyprod  
create directory vmsys:p735falk.tcpip.source
```

Only create the Source directory if it does not already exist.

### 6.5.3 Install the Kerberos US Feature

- 1 Log on the installation user ID, **P735FALK**.
- 2 Establish read access to the 5E5 minidisk and write access to the 51D minidisk (Software Inventory), if they are not already linked correctly.  
**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user who has it linked R/W link it as R/O. You then can issue the following commands to obtain R/W access to it.

```
link maint 51d 51d mr  
access 51d d
```

- 3 Have the Kerberos US Feature installation tape mounted and attached to P735FALK at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 4 Install the Kerberos US Feature.

**Notes:**

- If you've already created a PPF override file, you should specify your override file name after the **PPF** keyword for the following VMFINS command.
- You may be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you're unsure how to respond to a prompt, refer to the "Installing Products with VMFINS" and "Install Scenarios" chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.

```
vmfins install ppf 5735faln {vmkerb | vmkerbsfs} (nomemo nolink
```

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```

VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALN component VMKERB passed requisite checking
Do you want to create an override for 5735FALN VMKERB (prodid 5735FALN)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINT2760I VMFINST processing started
VMFLDP2706I 5735FALN VMKERB (prodid 5735FALN) will be processed as a PDI
product
VMFSET2760I VMFSETUP processing started
VMFUTL2205I Minidisk|Directory Assignments:
String Mode Stat Vdev Label/Directory
VMFUTL2205I LOCALMOD E R/W 2C4 TCP2C4
VMFUTL2205I APPLY F R/W 4A6 KER4A6
VMFUTL2205I G R/W 4A2 KER4A2
VMFUTL2205I DELTA H R/W 2D2 TCP2D2
VMFUTL2205I BUILD1 I R/W 491 TCP491
VMFUTL2205I BUILD3 J R/W 492 TCP492
VMFUTL2205I BASE1 K R/W 2B2 TCP2B2
VMFUTL2205I BASE2 L R/W 2B3 TCP2B3
VMFUTL2205I ----- A R/W 191 TCP191
VMFUTL2205I ----- B R/O 5E5 MNT5E5
VMFUTL2205I ----- D R/W 51D MNT51D
VMFUTL2205I ----- S R/O 190 MNT190
VMFUTL2205I ----- Y/S R/O 19E MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS TAPE 9430
VMFREC1851I (1 of 8) VMFRCAXL processing AXLIST
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (2 of 8) VMFRCPTF processing PARTLST
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (3 of 8) VMFRCOM processing DELTA
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (4 of 8) VMFRCALL processing APPLY
VMFRCX2159I Loading part(s) to APPLY 4A6 (F)
VMFRCX2159I Loaded nnn part(s) to APPLY 4A6 (F)
VMFREC1851I (5 of 8) VMFRCALL processing KERB
VMFRCX2159I Loading part(s) to BASE1 2B2 (K)
VMFRCX2159I Loaded nnn part(s) to BASE1 2B2 (K)
VMFREC1851I (6 of 8) VMFRCALL processing SERVER
VMFRCX2159I Loading part(s) to BUILD1 491 (I)
VMFRCX2159I Loaded nnn part(s) to BUILD1 491 (I)
VMFREC1851I (7 of 8) VMFRCALL processing CLIENT
VMFRCX2159I Loading part(s) to BUILD3 492 (J)
VMFRCX2159I Loaded nnn part(s) to BUILD3 492 (J)
VMFREC1851I (8 of 8) VMFRCALL processing SOURCE
VMFRCX2159I Loading part(s) to BASE2 2B3 (L)
VMFRCX2159I Loaded nnn part(s) to BASE2 2B3 (L)
VMFREC2760I VMFREC processing completed successfully
VMFINT2760I VMFINST processing completed successfully
VMFINS2760I VMFINS processing completed successfully

```

**5** Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview install**

### 6.5.3.1 Update Build Status Table for the Kerberos US Feature

- 1 Update the VM SYSBLDS software inventory file for the Kerberos US Feature.

**vmfins build ppf 5735faln {vmkerb | vmkerbsfs} (serviced nolink**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

The SERVICED option will build any parts that were not built on the installation tape and update the Software Inventory build status table showing that the product 5735FALN has been built.

## 6.5.4 Place the Kerberos US Feature Into Production

### 6.5.4.1 Copy Kerberos US Feature Files Into Production

- 1 Log on the installation user ID, **P735FALK**.
- 2 Establish read access to the VMSES/E code and the Software Inventory minidisks.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

**link maint 51d 51d rr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

- 3 Establish the correct minidisk access order.

**vmfsetup 5735faln {vmkerb | vmkerbsfs} (link**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

- 4 If necessary, modify the 5735FALN CATALOG file so that the appropriate files will be placed into production for your environment.

The 5735FALN CATALOG file is used by the TCP2PROD command to copy files to TCP/IP for VM minidisks. See Appendix A, "TCP2PROD" on page 172 for information about the TCP2PROD command and TCP/IP for VM catalog files.

**Notes:**

- a. You should modify both the product runtime file section (KERBRUN) and the customizable sample file (KERBCONFIG) section of the 5735FALN CATALOG at the same time. For reference purposes, files that can be processed using the KERBRUN section of this file are listed in Figure 35 on page 97; those processed using the KERBCONFIG section are listed in Figure 36 on page 98.
- b. To ensure you are notified of any service-related changes to the 5735FALN CATALOG file, changes should be made using a VMSES/E-format local modification. See Appendix B, "Modifying TCP/IP for VM CATALOG Files" on page 178 for more information about how to change 5735FALN CATALOG file in this manner.

**5** Establish a write link to any TCP/IP for VM production or server minidisks which are not yet linked in this mode.

If you did not add the appropriate LINK statements to the P735FALK directory when TCP/IP for VM resources were allocated (based on information in the 5735FALN PLANINFO file), you will need to manually link the TCP/IP production and server minidisks necessary for your environment. See Figure 35 on page 97 and Figure 36 on page 98 for the link addresses you should use. If you have a PPF override that has changed any of these addresses, use your values.

```
link tcpipid vdev1 vdev2 mr
```

**Note:** If another user already has a required minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the minidisk in read-only mode (RR), and then re-issue the above LINK command. Do not continue with these procedures until a R/W link is established for the minidisk in question.

**6** Copy Kerberos US Feature files into production using the TCP2PROD command. The files copied via the following command are those identified in the KERBRUN section of the 5735FALN CATALOG file.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735faln {vmkerb | vmkerbsfs} 5735faln kerbrun**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

#### **6.5.4.2 Configure the Kerberos US Feature**

The Kerberos US Feature has been installed with sample configuration files which are used by the Kerberos servers and clients. These files need to be copied to the appropriate minidisk, renamed if necessary, and customized for your installation. Use the TCP2PROD command, as described below, to copy these files into production.

See the *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for detailed explanations about how to configure the Kerberos servers.

**Note:** For step 2 below, it's assumed that the KERBCONFIG section of the 5735FALN CATALOG has been suitably modified, as described in step 4 of 6.5.4.1, "Copy Kerberos US Feature Files Into Production" on page 94. If this is not the case, you should make any necessary changes to the KERBCONFIG section of the 5735FALN CATALOG file before you continue with the following steps.

- 1** If necessary, establish the appropriate environment, as described by steps 1 through 3, in 6.5.4.1, "Copy Kerberos US Feature Files Into Production" on page 94.
- 2** Copy Kerberos US Feature configuration files into production using the TCP2PROD command. For reference purposes, files that can be processed using the KERBCONFIG section are listed in Figure 36 on page 98.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735faln {vmkerb | vmkerbsfs} 5735faln kerbconfig**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

For information about copying client code to the Product Code minidisk, see Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182.

**6.5.4.3 Kerberos US Feature Runtime and Sample Configuration Files**

Figure 35 lists the name and location of the Kerberos US Feature runtime files provided by IBM, and the names and location of these files after they've been placed into production.

*Figure 35 (Page 1 of 2). Files to Copy into Production - Kerberos US Feature*

<b>P735FALK Test Disk</b>	<b>P735FALK Link Address</b>	<b>Original File Name/Type</b>	<b>Production File Name/Type</b>	<b>Server Minidisk</b>
491	591	ADM_SERV MODULE	no change	_____
491	591	ADM_SPRO EXEC	no change	_____
491	591	EXT_SRVT MODULE	no change	_____
491	591	KADMEXIT EXEC	no change	_____
491	591	KDB_DEST MODULE	no change	_____
491	591	KDB_EDIT MODULE	no change	_____
491	591	KDB_INIT MODULE	no change	_____
491	591	KDB_UTIL MODULE	no change	_____
491	591	KERBEROS MODULE	no change	_____
491	591	KERBEXIT EXEC	no change	_____
491	591	VMKERPRO EXEC	no change	_____
491	276	VMKERPRO EXEC	PROFILE EXEC	VMKERB 191
491	277	ADM_SPRO EXEC	PROFILE EXEC	ADMSERV 191
492	592	BPLDBM TXTLIB	no change	_____
492	592	DES H	no change	_____
492	592	DES TXTLIB	no change	_____

*Figure 35 (Page 2 of 2). Files to Copy into Production - Kerberos US Feature*

<b>P735FALK Test Disk</b>	<b>P735FALK Link Address</b>	<b>Original File Name/Type</b>	<b>Production File Name/Type</b>	<b>Server Minidisk</b>
492	592	DES@EXT H	no change	_____
492	592	KADMIN MODULE	no change	_____
492	592	KDB TXTLIB	no change	_____
492	592	KDESTROY MODULE	no change	_____
492	592	KINIT MODULE	no change	_____
492	592	KLIST MODULE	no change	_____
492	592	KPASSWORD MODULE	no change	_____
492	592	KRB H	no change	_____
492	592	KRB TXTLIB	no change	_____
492	592	KRB@EXT H	no change	_____
492	592	KRBSAMP EXEC	no change	_____
492	592	KSTASH MODULE	no change	_____
492	592	LSB@ADDR H	no change	_____
492	592	MIT-COPY H	no change	_____
492	592	PROT H	no change	_____
492	592	SAMPLE_C LOADLIST	no change	_____
492	592	SAMPLE_C MODULE	no change	_____
492	592	SAMPLE_S LOADLIST	no change	_____
492	592	SAMPLE_S MODULE	no change	_____
492	592	SAMPLE@C C	no change	_____
492	592	SAMPLE@S C	no change	_____

Figure 36 lists the name and location of the Kerberos US Feature sample files as provided by IBM, and the default names and location of these files after they've been customized.

*Figure 36. Files to Customize - Kerberos US Feature*

<b>Sample Disk</b>	<b>Configured Disk</b>	<b>Sample File Name/Type</b>	<b>Configured File Name/Type</b>
591	198	KADMEXIT EXEC	no change
591	198	KERBEXIT EXEC	no change
2B2	592	KRB CONF	no change
2B2	277 (1*)	ADM@ACL ADD	no change
2B2	277 (1*)	ADM@ACL GET	no change
2B2	277 (1*)	ADM@ACL MOD	no change

**Notes:**

1. This address (277) is the P735FALK link address for the ADMSERV 191 minidisk.



#### 6.5.4.4 Unpack Source Files

The Kerberos US Feature has been installed with packed source files. If you wish to unpack the source files, log on P735FALK and invoke the copy command with the unpack option. The Source disk must have been defined large enough to store the unpacked files.

**access 2b3 m**  
**copyfile *fn ft* m (unpack olddate replace**

Access the Source directory if you installed to the Shared File System.

If **all** files on the disk are packed, you can unpack them all by specifying \* \* for *fn ft*; otherwise, you should unpack Kerberos US Feature files on an individual basis.

**The Kerberos US Feature is now installed and built on your system.**

---

## 6.6 Installing TCP/IP V2 R4 for VM - Kerberos WT Feature

### 6.6.1 Plan Your Installation for the Kerberos WT Feature

The VMFINS command will be used to plan the installation. This section has 2 main steps that will:

- load the first tape file, containing installation files
- generate a “PLANINFO” file listing
  - all user ID and MDISK (minidisk) requirements
  - required products

To obtain planning information for your environment:

- 1** Log on the installation user ID, **P735FALK**.
- 2** Mount the Kerberos WT Feature installation tape and attach it to the user ID at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 3** Establish read access to the VMSES/E code.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

- 4** Establish write access to the Software Inventory disk.

**link maint 51d 51d mr  
access 51d d**

The MAINT 51D minidisk is where the VMSES/E system-level Software Inventory and other dependent files reside.

**Note:** If another user already has the MAINT 51D minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the 51D in read-only mode (RR), and then re-issue the above LINK and ACCESS commands. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

- 5** Load the Kerberos WT Feature product control files to the 51D minidisk.

## vmfins install info (nomemo)

The NOMEMO option will load the memo from the tape but will not issue a prompt to send them to the system printer. Specify the MEMO option if you want to be prompted for printing the memo.

This command will perform the following:

- load Memo-to-Users
- load various product control files, including the Product Parameter File (PPF) and the PRODPART files
- create VMFINS PRODLIST on your A-disk. The VMFINS PRODLIST contains a list of products on the installation tape.

```
VMFINS2760I VMFINS processing started
VMFINS1909I VMFINS PRODLIST created on your A-disk
VMFINS2760I VMFINS processing completed successfully
Ready;
```

## 6 Obtain resource planning information for the Kerberos WT Feature.

**Note:** The product will **not** be loaded by the VMFINS command at this time.

## vmfins install ppf 5735falp {vmkerb | vmkerbsfs} (plan nomemo)

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product to minidisks or **vmkerbsfs** if you installed it to Shared File System directories.

The PLAN option indicates that VMFINS will perform requisite checking, plan system resources, and provide an opportunity to override the defaults in the product parameter file.

### You can override any of the following:

- the name of the product parameter file
- the default user IDs
- minidisk/directory definitions

**Notes:**

- a. If you change the PPF name, a default user ID, or other parameters via a PPF override, you'll need to use your changed values instead of those indicated (when appropriate), throughout the rest of the installation instructions, as well as those provided for servicing Kerberos US Feature. For example, you'll need to specify your PPF override file name instead of 5735FALN for certain VMSES/E commands.
- b. If you're not familiar with creating PPF overrides using VMFINS, you should review the "Using the Make Override Panel" section in Chapter 3 of the *VMSES/E Introduction and Reference* before you continue.
- c. For more information about changing the VMSYS file pool name, refer to Chapter 3 of the *VMSES/E Introduction and Reference*.

```
VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALP component VMKERB passed requisite checking
Do you want to create an override for 5735FALP VMKERB (prodid 5735FALP)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFRMT2760I VMFRMT processing started
VMFRMT2760I VMFRMT processing completed successfully
VMFPLA1909I 5735FALP PLANINFO created on your A-disk
VMFINS2760I VMFINS processing completed successfully
```

**Note!**

**Complete this step only if you received message VMFREQ2806W.**

If you receive the following VMFREQ2806W message (specifying either VMKERB or VMKERBSFS), you must do some additional processing.

```
VMFREQ2806W The following requisites for product 5735FALP component VMKERB are
              not satisfied
VMFREQ2806W Type          Product  Component      PTF
              -----
VMFREQ2806W Prerequisite  5735FALK
VMFREQ2806W Requisite    5688198E
VMFREQ2806W OR           2VMVMA10
```

The messages indicate that 5735FALK, product TCP/IP V2 R4 for VM, is a prerequisite product for the Kerberos WT Feature. You must install it before you can continue with this installation.

In addition, the messages indicate that 5688198E, the IBM Language Environment for MVS and VM Release 5 product, is a requisite. You can install the NFS Feature without this requisite, but will have to install IBM Language Environment for MVS and VM Release 5 before you run or service the Kerberos WT Feature.

If you are going to install IBM Language Environment for MVS and VM Release 5, you can continue with installing Kerberos WT Feature. IBM Language Environment for MVS and VM Release 5 is in VMSES/E format, so the VMSES/E system inventory will be updated automatically when you install this requisite product.

- 7 Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview install

### 6.6.2 Allocate Resources for Installing the Kerberos WT Feature

Use the planning information in the 5735FALP PLANINFO file, created in the **PLAN** step, to:

- Update the P735FALK user directory entry for minidisk install

**OR**

Update the P735FALK user directory entry for SFS install

- Create the other required user directory entries.

### 6.6.2.1 PROFILE Directory Entries

There is a PROFILE directory entry, PROFILE CMSUSER, that is used in the new directory entries. This profile should already have been added to the system directory when you installed the TCP/IP V2 R4 for VM base product. The profile is as follows:

```
PROFILE CMSUSER
  IPL CMS
  MACHINE XA
  SPOOL 00C 2540 READER *
  SPOOL 00D 2540 PUNCH A
  SPOOL 00E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 190 190 RR
  LINK MAINT 19D 19D RR
  LINK MAINT 19E 19E RR
```

### 6.6.2.2 Installing the Kerberos WT Feature to Minidisks

**1** Obtain the user directory from the 5735FALP PLANINFO file.

**Note:** The user directory entries are located at the end of the resource section at the bottom of the PLANINFO file. These entries contain all of the links and privilege classes necessary for the Kerberos WT Feature user IDs. Use the directory entries found in the PLANINFO file as models for input to your system directory.

**2** Modify the directory entries, using Figure 29 on page 36 and Figure 30 on page 37 to obtain the minidisk requirements.

- Add the MDISK statements to the directory entries for P735FALK, VMKERB, and ADMSERV. These user IDs are required. Also add an MDISK statement to P735FALK for the Source disk if it does not already exist. Use one of the two Source disk sizes specified in Figure 30 on page 37 depending on whether you intend to unpack the source files after installation.

- Add the following links to the P735FALK directory entry

```
LINK VMKERB 191 276 MR
LINK ADMSERV 191 277 MR
```

- Increase the size of the existing P735FALK and TCPMAINT service disks by the amount specified in Figure 30 on page 37 if this has not already been done. Use one of the two Source disk sizes, depending on whether you intend to unpack the source files after installation.

**3** Update the system directory

- Add the directory entries for VMKORB and ADMSERV. Change the passwords from xxxxx to a valid password, in accordance with your security guidelines.
- Modify the directory entries for P735FALK and TCPMAINT.

Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

All new minidisks must be formatted before you install the Kerberos WT Feature.

### 6.6.2.3 Installing the Kerberos WT Feature to SFS Directories

- 1 Obtain the user directory from the 5735FALP PLANINFO file.

**Note:** The user directory entries are located at the end of the resource section at the bottom of the PLANINFO file. These entries contain all of the links and privilege classes necessary for the Kerberos WT Feature user IDs. Use the directory entries found in the PLANINFO file as models for input to your system directory.

- 2 Modify the directory entries, using Figure 29 on page 36 and Figure 30 on page 37 to obtain the minidisk requirements for the minidisks that do not have Shared File System directories specified.

- Add the MDISK statements to the directory entries for VMKORB and ADMSERV. These user IDs are required.
- Add the following links to the P735FALK directory entry

```
LINK VMKORB 191 276 MR
LINK ADMSERV 191 277 MR
```

- Increase the size of the existing P735FALK and TCPMAINT service disks that are not on the Shared File System by the amount specified in Figure 30 on page 37 if this has not already been done.

- 3 Update the system directory

- Add the directory entries for VMKORB and ADMSERV. Change the passwords from xxxxx to a valid password, in accordance with your security guidelines.
- Modify the directory entries for P735FALK and TCPMAINT.

Place the new directory on-line using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

**Note**

All new minidisks must be formatted before you install the Kerberos WT Feature.

**4** An SFS install will also require the following steps:

- a** Determine the number of 4k blocks that are required for SFS directories by adding up the 4K blocks required for each SFS directory you plan to use. Use one of the two Source disk sizes, depending on whether you intend to unpack the source files after installation.

If you intend to use all of the default Kerberos WT Feature SFS directories, the 4K block requirements are summarized in Figure 29 on page 36 and Figure 30 on page 37. This information will be used to modify the storage allocation for P735FALK in the VMSYS filepool.

- b** Increase the amount of SFS storage allocated to user P735FALK in the VMSYS filepool, if sufficient storage was not already allocated. Use the MODIFY USER command:

```
MODIFY USER +blocks FOR P735FALK VMSYS:
```

where *blocks* is the number of 4k blocks that you calculated in the previous step.

**Note:** This must be done from a user ID that is an administrator for VMSYS: filepool.

- c** Determine if there are enough blocks available in the filepool to install the Kerberos WT Feature. This information can be obtained from the QUERY FILEPOOL STATUS command. Near the end of the output from this command is a list of minidisks in the filepool and the number of blocks free. If the number of blocks free is smaller than the total 4k blocks needed to install the Kerberos WT Feature you will need to add space to the filepool. See *VM/ESA File Pool Planning, Administration and Operation* or *VM/ESA SFS and CRR Planning, Administration and Operation* for information about adding space to a filepool.

- d** Create the necessary subdirectories listed in the 5735FALP PLANINFO file using the CREATE DIRECTORY command. If necessary, see the *VM/ESA CMS Command Reference* for more information about the CREATE DIRECTORY command.



```
set filepool vmsys:  
create directory vmsys:p735falk.vmkerb  
create directory vmsys:p735falk.vmkerb.applyalt  
create directory vmsys:p735falk.vmkerb.applyprod  
create directory vmsys:p735falk.tcpip.source
```

Only create the Source directory if it does not already exist.

### 6.6.3 Install the Kerberos WT Feature

- 1 Log on the installation user ID, **P735FALK**.
- 2 Establish read access to the 5E5 minidisk and write access to the 51D minidisk (Software Inventory), if they are not already linked correctly.  
**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user who has it linked R/W link it as R/O. You then can issue the following commands to obtain R/W access to it.

```
link maint 51d 51d mr  
access 51d d
```

- 3 Have the Kerberos WT Feature installation tape mounted and attached to P735FALK at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 4 Install the Kerberos WT Feature.

**Notes:**

- If you've already created a PPF override file, you should specify your override file name after the **PPF** keyword for the following VMFINS command.
- You may be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you're unsure how to respond to a prompt, refer to the "Installing Products with VMFINS" and "Install Scenarios" chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.

```
vmfins install ppf 5735falp {vmkerb | vmkerbsfs} (nomemo nolink
```

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product to minidisks or **vmkerbsfs** if you installed it to Shared File System directories.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```

VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALP component VMKERB passed requisite checking
Do you want to create an override for 5735FALP VMKERB (prodid 5735FALP)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINT2760I VMFINST processing started
VMFLDP2706I 5735FALP VMKERB (prodid 5735FALP) will be processed as a PDI
product
VMFSET2760I VMFSETUP processing started
VMFUTL2205I Minidisk|Directory Assignments:
String Mode Stat Vdev Label/Directory
VMFUTL2205I LOCALMOD E R/W 2C4 TCP2C4
VMFUTL2205I APPLY F R/W 4A6 KER4A6
VMFUTL2205I G R/W 4A2 KER4A2
VMFUTL2205I DELTA H R/W 2D2 TCP2D2
VMFUTL2205I BUILD1 I R/W 491 TCP491
VMFUTL2205I BUILD3 J R/W 492 TCP492
VMFUTL2205I BASE1 K R/W 2B2 TCP2B2
VMFUTL2205I BASE2 L R/W 2B3 TCP2B3
VMFUTL2205I ----- A R/W 191 TCP191
VMFUTL2205I ----- B R/O 5E5 MNT5E5
VMFUTL2205I ----- D R/W 51D MNT51D
VMFUTL2205I ----- S R/O 190 MNT190
VMFUTL2205I ----- Y/S R/O 19E MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS TAPE 9440
VMFREC1851I (1 of 8) VMFRCAXL processing AXLIST
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (2 of 8) VMFRCPTF processing PARTLST
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (3 of 8) VMFRCOM processing DELTA
VMFRCX2159I Loading nnn part(s) to DELTA 2D2 (H)
VMFREC1851I (4 of 8) VMFRCALL processing APPLY
VMFRCAX2159I Loading part(s) to APPLY 4A6 (F)
VMFRCAX2159I Loaded nnn part(s) to APPLY 4A6 (F)
VMFREC1851I (5 of 8) VMFRCALL processing KERB
VMFRCAX2159I Loading part(s) to BASE1 2B2 (K)
VMFRCAX2159I Loaded nnn part(s) to BASE1 2B2 (K)
VMFREC1851I (6 of 8) VMFRCALL processing SERVER
VMFRCAX2159I Loading part(s) to BUILD1 491 (I)
VMFRCAX2159I Loaded nnn part(s) to BUILD1 491 (I)
VMFREC1851I (7 of 8) VMFRCALL processing CLIENT
VMFRCAX2159I Loading part(s) to BUILD3 492 (J)
VMFRCAX2159I Loaded nnn part(s) to BUILD1 491 (I)
VMFREC1851I (8 of 8) VMFRCALL processing SOURCE
VMFRCAX2159I Loading part(s) to BASE2 2B3 (L)
VMFRCAX2159I Loaded nnn part(s) to BASE2 2B3 (L)
VMFREC2760I VMFREC processing completed successfully
VMFINT2760I VMFINST processing completed successfully
VMFINS2760I VMFINS processing completed successfully

```

**5** Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview install**

### 6.6.3.1 Update Build Status Table for the Kerberos WT Feature

- 1 Update the VM SYSBLDS software inventory file for the Kerberos WT Feature.

**vmfins build ppf 5735falp {vmkerb | vmkerbsfs} (serviced nolink**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product to minidisks or **vmkerbsfs** if you installed it to Shared File System directories.

The SERVICED option will build any parts that were not built on the installation tape and update the Software Inventory build status table showing that the product 5735FALP has been built.

## 6.6.4 Place the Kerberos WT Feature Into Production

### 6.6.4.1 Copy Kerberos WT Feature Files Into Production

- 1 Log on the installation user ID, **P735FALK**.
- 2 Establish read access to the VMSES/E code and the Software Inventory minidisks.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

**link maint 51d 51d rr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

- 3 Establish the correct minidisk access order.

**vmfsetup 5735falp {vmkerb | vmkerbsfs} (link**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

- 4 If necessary, modify the 5735FALP CATALOG file so that the appropriate files will be placed into production for your environment.

The 5735FALP CATALOG file is used by the TCP2PROD command to copy files to TCP/IP for VM minidisks. See Appendix A, “TCP2PROD” on page 172 for information about the TCP2PROD command and TCP/IP for VM catalog files.

**Notes:**

- a. You should modify both the product runtime file section (KERBRUN) and the customizable sample file (KERBCONFIG) section of the 5735FALP CATALOG at the same time. For reference purposes, files that can be processed using the KERBRUN section of this file are listed in Figure 37 on page 113; those processed using the KERBCONFIG section are listed in Figure 38 on page 114.
- b. To ensure you are notified of any service-related changes to the 5735FALP CATALOG file, changes should be made using a VMSES/E-format local modification. See Appendix B, “Modifying TCP/IP for VM CATALOG Files” on page 178 for more information about how to change 5735FALP CATALOG file in this manner.

**5** Establish a write link to any TCP/IP for VM production or server minidisks which are not yet linked in this mode.

If you did not add the appropriate LINK statements to the P735FALK directory when TCP/IP for VM resources were allocated (based on information in the 5735FALP PLANINFO file), you will need to manually link the TCP/IP production and server minidisks necessary for your environment. See Figure 37 on page 113 and Figure 38 on page 114 for the link addresses you should use. If you have a PPF override that has changed any of these addresses, use your values.

```
link tcpipid vdev1 vdev2 mr
```

**Note:** If another user already has a required minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the minidisk in read-only mode (RR), and then re-issue the above LINK command. Do not continue with these procedures until a R/W link is established for the minidisk in question.

**6** Copy Kerberos WT Feature files into production using the TCP2PROD command. The files copied via the following command are those identified in the KERBRUN section of the 5735FALP CATALOG file.

#### Note

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735falp {vmkerb | vmkerbsfs} 5735falp kerbrun**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

#### 6.6.4.2 Configure the Kerberos WT Feature

The Kerberos WT Feature has been installed with sample configuration files which are used by the Kerberos servers and clients. These files need to be copied to the appropriate minidisk, renamed if necessary, and customized for your installation. Use the TCP2PROD command, as described below, to copy these files into production.

See the *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for detailed explanations about how to configure the Kerberos servers.

**Note:** For step 2 below, it's assumed that the KERBCONFIG section of the 5735FALP CATALOG has been suitably modified, as described in step 4 of 6.6.4.1, "Copy Kerberos WT Feature Files Into Production" on page 110. If this is not the case, you should make any necessary changes to the KERBCONFIG section of the 5735FALP CATALOG file before you continue with the following steps.

- 1** If necessary, establish the appropriate environment, as described by steps 1 through 3, in 6.6.4.1, "Copy Kerberos WT Feature Files Into Production" on page 110.
- 2** Copy Kerberos US Feature configuration files into production using the TCP2PROD command. For reference purposes, files that can be processed using the KERBCONFIG section are listed in Figure 38 on page 114.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735falp {vmkerb | vmkerbsfs} 5735falp kerbconfig**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

For information about copying client code to the Product Code minidisk, see Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182.

### 6.6.4.3 Kerberos WT Feature Runtime and Sample Configuration Files

Figure 37 lists the name and location of the Kerberos WT Feature runtime files provided by IBM, and the names and location of these files after they've been placed into production.

<i>Figure 37 (Page 1 of 2). Files to Copy into Production - Kerberos WT Feature</i>				
<b>P735FALK Test Disk</b>	<b>P735FALK Link Address</b>	<b>Original File Name/Type</b>	<b>Production File Name/Type</b>	<b>Server Minidisk</b>
491	591	ADM_SERV MODULE	no change	_____
491	591	ADM_SPRO EXEC	no change	_____
491	591	EXT_SRVT MODULE	no change	_____
491	591	KADMEXIT EXEC	no change	_____
491	591	KDB_DEST MODULE	no change	_____
491	591	KDB_EDIT MODULE	no change	_____
491	591	KDB_INIT MODULE	no change	_____
491	591	KDB_UTIL MODULE	no change	_____
491	591	KERBEROS MODULE	no change	_____
491	591	KERBEXIT EXEC	no change	_____
491	591	VMKERPRO EXEC	no change	_____
491	276	VMKERPRO EXEC	PROFILE EXEC	VMKERB 191
491	277	ADM_SPRO EXEC	PROFILE EXEC	ADMSERV 191
492	592	BPLDBM TXTLIB	no change	_____
492	592	DES H	no change	_____
492	592	DES@EXT H	no change	_____

<i>Figure 37 (Page 2 of 2). Files to Copy into Production - Kerberos WT Feature</i>				
<b>P735FALK Test Disk</b>	<b>P735FALK Link Address</b>	<b>Original File Name/Type</b>	<b>Production File Name/Type</b>	<b>Server Minidisk</b>
492	592	KADMIN MODULE	no change	_____
492	592	KDB TXTLIB	no change	_____
492	592	KDESTROY MODULE	no change	_____
492	592	KINIT MODULE	no change	_____
492	592	KLIST MODULE	no change	_____
492	592	KPASSWORD MODULE	no change	_____
492	592	KRB H	no change	_____
492	592	KRB TXTLIB	no change	_____
492	592	KRB@EXT H	no change	_____
492	592	KRBSAMP EXEC	no change	_____
492	592	KSTASH MODULE	no change	_____
492	592	LSB@ADDR H	no change	_____
492	592	MIT-COPY H	no change	_____
492	592	PROT H	no change	_____
492	592	SAMPLE_C LOADLIST	no change	_____
492	592	SAMPLE_C MODULE	no change	_____
492	592	SAMPLE_S LOADLIST	no change	_____
492	592	SAMPLE_S MODULE	no change	_____
492	592	SAMPLE@C C	no change	_____
492	592	SAMPLE@S C	no change	_____

Figure 38 lists the name and location of the Kerberos WT Feature sample files as provided by IBM, and the default names and location of these files after they've been customized.

<i>Figure 38. Files to Customize - Kerberos WT Feature</i>			
<b>Sample Disk</b>	<b>Configured Disk</b>	<b>Sample File Name/Type</b>	<b>Configured File Name/Type</b>
591	198	KADMEXIT EXEC	no change
591	198	KERBEXIT EXEC	no change
2B2	592	KRB CONF	no change
2B2	277 (1*)	ADM@ACL ADD	no change
2B2	277 (1*)	ADM@ACL GET	no change
2B2	277 (1*)	ADM@ACL MOD	no change
<b>Notes:</b>			
1. This address (277) is the P735FALK link address for the ADMSERV 191 minidisk.			



**Note:** The Configured Disk address 277 in Figure 38 corresponds to the ADMSERV 191 minidisk.

#### 6.6.4.4 Unpack Source Files

The Kerberos WT Feature has been installed with packed source files. If you wish to unpack the source files, log on P735FALK and invoke the copy command with the unpack option. The Source disk must have been defined large enough to store the unpacked files.

**access 2b3 e**  
**copyfile *fn ft* e (unpack olddate replace**

Access the Source directory if you installed to the Shared File System.

If **all** files on the disk are packed, you can unpack them all by specifying \* \* for *fn ft*; otherwise, you should unpack Kerberos WT Feature files on an individual basis.

**The Kerberos WT Feature is now installed and built on your system.**

---

## 7.0 Service Instructions

This section of the Program Directory contains the procedures to install CORrective and preventive service to the TCP/IP V2 R4 for VM base product and the NFS, Kerberos US, and Kerberos World Trade features, using VMSES/E. Service for the Source feature is included in the service for the base product. Preventive service is delivered via a Recommended Service Upgrade (RSU) tape.

To become more familiar with service using VMSES/E, you should read the introductory chapters in:

- *VMSES/E Introduction and Reference* (SC24-5747 or SC24-5444)

These manuals also contain the command syntax for the VMSES/E commands listed in the procedure.

Each step of the servicing instructions must be followed. Do not skip any step unless otherwise directed. All instructions showing accessing of disks assume the use of default minidisk addresses. If different minidisk addresses are used, or if using the Shared File System, change the instructions appropriately.

**Note!**

The sample console output presented throughout these instructions was produced on a VM/ESA Version 2 Release 2.0 system and assumes that you installed to minidisks using the default PPF and component names. If you installed the TCP/IP V2 R4 for VM base product and features on a different VM/ESA system or used a different PPF or component, the results obtained for some commands may differ from those depicted here.

---

### 7.1 VMSES/E Service Process Overview

The following is a brief description of the main steps in servicing TCP/IP V2 R4 for VM and the features using VMSES/E.

- **Merging Service**  
Use the VMFMRDSK command to clear the alternate apply disk before receiving new service. This allows you to easily remove the new service if a serious problem is found.
- **Receiving Service**  
The VMFREC command receives service from the delivery media and places it on the Delta disk.
- **Applying Service**  
The VMFAPPLY command updates the version vector table (VVT), which identifies the service level of all the serviced parts. In addition, AUX files are generated from the VVT for parts that require them.
- **Reapplying Local Service** (if applicable)

All local service (mods) must be entered into the software inventory to allow VMSES/E to track the changes and build them into the system. Refer to Chapter 7 of the *VM/ESA Service Guide* for this procedure.

- Building New Levels

The build task generates the serviced level of an object and places the new object on a test BUILD disk.

- Placing the New Service into Production

Once the service is satisfactorily tested it should be put into production by copying the new service to the production disks. Customized files that have been serviced must be customized again.

---

## 7.2 Servicing TCP/IP V2 R4 for VM

### 7.2.1 Important Service Notes

#### 7.2.1.1 PPF Override Considerations

- The *ppfname* used throughout these servicing instructions is **5735FALK**, which assumes you are using the PPF supplied by IBM for the TCP/IP V2 R4 for VM. If you have your own PPF override file for the TCP/IP V2 R4 for VM, you should use your file's *ppfname* instead of **5735FALK**. The *ppfname* you use should be used **throughout** the rest of this procedure.

#### 7.2.1.2 Language Environment Runtime Library Considerations

- If TCP/IP V2 R4 for VM is installed on VM/ESA Version 1 Release 2.2, IBM Language Environment for MVS and VM Release 5 (5688198) is required to service several modules used for certain TCP/IP for VM functions (see 5.2.2, "Other Program Product Requirements" on page 17 for a list of such functions). For VM/ESA Version 2 Release 1.0 or later, this requirement is met by the Common Execution Library (CEL) included with these releases of VM/ESA.

If you do not have Language Environment runtime library support installed, you must use one of the following component names when you service TCP/IP for VM:

**tcpipnoc** — if installed to minidisks

**tcpipsfsnoc** — if installed to Shared File System

Using one of these names will avoid building the C components of TCP/IP for VM. As stated above, use your own override for the appropriate component if you created one.

- If you service TCP/IP for VM C components, the Language Environment runtime library must be available when you build serviced objects using the VMFBLD command. If the Language Environment runtime library does not reside on a system minidisk automatically accessed by VMSES/E (such as the MAINT 19E minidisk), you need to ensure the appropriate minidisk is available (perhaps through the use of a PPF override).

## Quick Index for Service Instructions

Select the service instructions you should use, based on the type of service you are installing:

- TCP/IP for VM RSU Service — Begin with 7.2.2, “Preventive Service for TCP/IP V2 R4 for VM” on page 119
- TCP/IP for VM COR Service — Begin with 7.2.3, “Corrective Service for TCP/IP V2 R4 for VM” on page 126
- NFS Feature RSU Service — Begin with 7.4.2, “Preventive Service for NFS Feature (RSU)” on page 138
- NFS Feature COR Service — Begin with 7.4.4, “Corrective Service for NFS Feature” on page 144
- Kerberos US Feature COR Service — Begin with 7.5.2, “Corrective Service for Kerberos US Feature” on page 154
- Kerberos WT Feature COR Service — Begin with 7.6.2, “Corrective Service for Kerberos WT Feature” on page 163

## 7.2.2 Preventive Service for TCP/IP V2 R4 for VM

Preventive service is available periodically on the Recommended Service Upgrade (RSU) tape. Each RSU is cumulative and contains selected important PTFs. The service on the RSU is in pre-applied, pre-built format, and includes the serviced files, the objects that were rebuilt using these files, and the updated software inventory. This makes installing the new service much quicker, but will require that you re-apply any reach-ahead service (service that you have applied that is not on the RSU). RSU tapes are in install format, so the VMFINS command is used to load the tape.

### 7.2.2.1 Prepare to Receive Service

**1** Log on the TCP/IP V2 R4 for VM service user ID, **P735FALK**.

**2** Establish access to the software inventory disk.

**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user that has it accessed R/W link it R/O. You then can issue the following commands to obtain R/W access to it.

**link maint 51d 51d mr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Have the TCP/IP V2 R4 for VM RSU tape mounted and attached as 181 to **P735FALK**.

- 4 Receive the product documentation (5735FALK MEMO) to the 51D minidisk and identify the products and components that have service on the tape. The product documentation contains the amount of storage needed to receive the service on the RSU. Use this to ensure that there is enough space on the service disks or directories.

**vmfins install info (nomemo**

**nomemo** will load but not print the memos.

- 5 Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

**vmfmrdsk 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} apply (setup**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

This command copies the alternate APPLY disk to the production APPLY disk and then clears the alternate APPLY disk.

- 6 Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview mrd**

- 7 Invoke the VMFPSU command to obtain additional information about the service contained on the RSU and how it will affect your local modifications. This command creates an output file, *appid PSUPLAN*, which you can review. See the *VM/ESA Service Guide* for an explanation of this file.

**vmfpsu 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc}**

Use **tcPIP** or **tcPIPSfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcPIPnoc** (minidisks) or **tcPIPSfsnoc** (SFS). **tcPIPnoc** (minidisks) or **tcPIPSfsnoc** (SFS).

This command produces an output file comparing the service on the RSU to the service on your system. The file name is *appid* **PSUPLAN**, where *appid* is specified in the PPF file.

### 7.2.2.2 Receive the Service

#### 1 Receive the service on the RSU.

Since the RSU contains pre-applied, pre-built service in install format, the VMFINS command will load the new service to the DELTA disk, the updated apply service inventory to the APPLY disk, and the pre-built objects to the appropriate test build disks.

**vmfins install ppf 5735falk {tcPIP | tcPIPSfs | tcPIPnoc | tcPIPSfsnoc} (nomemo nolink**

Use **tcPIP** or **tcPIPSfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcPIPnoc** (minidisks) or **tcPIPSfsnoc** (SFS).

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```
VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALK component TCPIP passed requisite checking
Do you want to create an override for 5735FALK TCPIP (prodid 5735FALK)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
.
.
.
VMFINS2760I VMFINS processing completed successfully
```

- 2 Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview install

### 7.2.2.3 Apply the Service

Since the service on the RSU is already pre-applied, this step will just reapply any reach-ahead service (service that is on your system but not on the RSU).

- 1 Reapply reach-ahead service.

## vmfapply ppf 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc}

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

This command reapplies the reach-ahead service on your system. The version vector table (VVT) is updated with all serviced parts and all necessary AUX files are generated on the alternate apply disk.

- 2 Review the apply message log (\$VMFAPP \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview apply

- 3 Re-work and re-apply local service, if it has been affected by the service on the RSU.
  - a. Affected local modifications are indicated in the output file created by the VMFPSU command invoked in a step 7 on page 120.
  - b. For information on re-working local modifications, refer to Chapter 7 in the *VM/ESA Service Guide*. Follow the steps that are applicable to your local modification.

The following substitutions may need to be made:

- **esalcl** should be **5735falk**
- **esa** should be **5735falk**



- *compname* should be **tcpip**, **tcpiptsfs**, **tcpiptnoc**, or **tcpiptsfnoc**
- *appid* should be **5735falk**
- *fm-local* should be the fm of 2C4
- *fm-applyalt* should be the fm of 2A6

Keep in mind that when you get to the following step in the *VM/ESA Service Guide*:

- "Rebuilding Objects" (VM/ESA Version 2 Release 1.0, or later)
- "Return to the Appropriate Section to Build Remaining Objects" (VM/ESA Version 1 Release 2.2)

you should return back to this program directory and continue with 7.2.2.4, "Update the Build Status Table."

#### 7.2.2.4 Update the Build Status Table

- 1 Update the Build Status Table with serviced parts.

**vmfbld ppf 5735falk {tcpip | tcpiptsfs | tcpiptnoc | tcpiptsfnoc} (status**

Use **tcpip** or **tcpiptsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpiptnoc** (minidisks) or **tcpiptsfnoc** (SFS).

This command updates the Build Status Table to determine what remains to be built.

**Note**

If a \$PPF file has been serviced you will get the following prompt:

```
VMFBLD2185R The following source product parameter files have been
              serviced:
VMFBLD2185R 5735FALK $PPF
VMFBLD2185R When source product parameter files are serviced, all
              product parameter files built from them must be recompiled
              using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
              parameter files built to your A-disk and exit VMFBLD so
              you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
              recompiled your product parameter files with VMFPPF.
```

**0**

Enter a 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 5735FALK $PPF
              on 191 (A) from level $PFnnnnn
```

**vmfppf 5735falk \***

**Note:** If you've created your own PPF override, use your PPF name instead of 5735FALK.

**copyfile 5735falk \$ppf a = = d (olddate replace  
erase 5735falk \$ppf a**

**Note:** Do not use your own PPF name in place of 5735FALK for the COPYFILE and ERASE commands.

Note Continued ...

**vmfbld ppf 5735falk {tcpip | tcpipsfs| tcpipnoc | tcpipsfsnoc} (status**

**1**

Re-issue VMFBLD to complete updating the build status table.

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS). When you receive the prompt that was previously displayed, enter a 1 to continue.

- 2** Use VMFVIEW to review the build status messages, and see what objects need to be built.

**vmfview build**

### 7.2.2.5 Build Serviced Objects

The RSU contains pre-built service, so this step will build only objects that were affected by any reach-ahead and local service that was reapplied.

- 1** Rebuild TCP/IP V2 R4 for VM serviced parts.

**vmfbld ppf 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} (serviced**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

- 2** Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview build**

### 7.2.2.6 Test the New Service

You should thoroughly test the service before putting it into production. To do this, add links in each server machine that you are using to link the test build disks for the server and client code instead of the production build disks. You will have to detach the 591 and 592 disks first. The link commands to issue are:

```
LINK P735FALK 491 591 RR
LINK P735FALK 492 592 RR
```

In addition, the NCS server virtual machine should link the test NCS administration code if it was serviced. Detach the 195 disk first.

```
LINK P735FALK 395 195 RR
```

### 7.2.2.7 Place the Service into Production

When you have thoroughly tested the new service, you should put the service into production. See 7.2.4, "Place the New TCP/IP V2 R4 for VM Service Into Production" on page 133 for an explanation of this task.

### 7.2.2.8 Determine RSU Service Level

The service contained on each RSU constitutes a new service level. Use this service level when ordering corrective service. The service level is updated in the system inventory when the RSU is installed.

The following command is used to query the current service level of the system.

```
vmfsim query vm sysrecs tdata :ppf 5735falk :stat
```

```
VMFSIP2408I RESULTS FOR
      TDATA :PPF 5735FALK :STAT
:PPF 5735FALK TCP/IP
:STAT RECEIVED.06/28/96.10:34:12.P735FALK.210-9602
```

The last part of the status line indicates the RSU service level: 9602.

## 7.2.3 Corrective Service for TCP/IP V2 R4 for VM

Corrective service for TCP/IP V2 R4 for VM is provided in COR format via tape or electronic envelope. It is installed using VMFREC, VMFAPPLY, and VMFBLD.

### 7.2.3.1 Prepare to Receive Service

**1** Log on the TCP/IP V2 R4 for VM service user ID, **P735FALK**.

**2** Establish access to the software inventory disk.

**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user that has it accessed R/W link it R/O. You then can issue the following commands to obtain R/W access to it.

**link maint 51d 51d mr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Have the TCP/IP V2 R4 for VM CORrective service tape mounted and attached as 181 to **P735FALK**.

**4** Establish the correct minidisk access order.

**vmfsetup 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc}**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

**5** Receive the documentation. VMFREC, with the INFO option, loads the documentation and displays a list of all the products on the tape.

#### Electronic Service

If you are receiving service from ServiceLink (electronic service) see Appendix A, "Receiving Service for VMSES Envelopes," section Receive Service Documentation, in the *VM/ESA Service Guide*. Then continue with step 7 on page 128.

**vmfrec info**

This command will load the service memo to the 191 disk.

**6** Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview receive

Also make note of which products and components have service on the tape. To do this, use the PF5 key to show all status messages which identify the products on the tape.

- 7 Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

## vmfmrdsk 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} apply

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

This command copies the alternate APPLY disk to the production APPLY disk and then clears the alternate APPLY disk.

- 8 Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview mrd

- 9 If you are installing corrective service for IBM Network Station support, see E.2.7.2, "Pre-Service Processing" on page 194 to determine whether additional preparation is necessary before you receive this service.

### 7.2.3.2 Receive the Service

#### Electronic Service

If you are receiving service from ServiceLink (electronic service) see Appendix A, "Receiving Service for VMSES Envelopes," in the *VM/ESA Service Guide*. Then continue with 7.2.3.3, "Apply the Service" on page 129.

- 1 Receive the service.

## vmfrec ppf 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc}

Use **tcpip** or **tcpiptsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpiptsfnoc** (SFS).

This command receives service from your service tape. All new service is loaded to the DELTA disk.

- 2** Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview receive**

### 7.2.3.3 Apply the Service

- 1** Apply the new service.

**vmfapply ppf 5735falk {tcpip | tcpiptsfs | tcpipnoc | tcpiptsfnoc}**

Use **tcpip** or **tcpiptsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpiptsfnoc** (SFS).

This command applies the service that you just received. The version vector table (VVT) is updated with all serviced parts and all necessary AUX files are generated on the alternate apply disk.

- 2** Review the apply message log (\$VMFAPP \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview apply**

#### Note

If you receive message VMFAPP2120W, you need to re-apply any local modifications before building the new TCP/IP V2 R4 for VM. Refer to chapter 7 in the *VM/ESA Service Guide*. Follow the steps that are applicable to your local modification.

The following substitutions may need to be made:

- **esalcl** should be **5735falk**
- **esa** should be **5735falk**
- *compname* should be **tcpip**, **tcpiptsfs**, **tcpipnoc**, or **tcpiptsfsnoc**
- *appid* should be **5735falk**
- *fm-local* should be the fm of 2C4
- *fm-applyalt* should be the fm of 2A6

Keep in mind that when you get to the following step in the *VM/ESA Service Guide*:

- "Rebuilding Objects" (VM/ESA Version 2 Release 1.0, or later)
- "Return to the Appropriate Section to Build Remaining Objects" (VM/ESA Version 1 Release 2.2)

you should return back to this program directory and continue with 7.2.3.4, "Update the Build Status Table" on page 130.

### 7.2.3.4 Update the Build Status Table

- 1 Update the Build Status Table with serviced parts.

**vmfbld ppf 5735falk {tcpip | tcpiptsfs | tcpipnoc | tcpiptsfsnoc} (status**

Use **tcpip** or **tcpiptsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpiptsfsnoc** (SFS).

This command updates the Build Status Table to include objects that have to be built due to the new service just applied.



**Note**

If a \$PPF file has been serviced you will get the following prompt:

```
VMFBLD2185R The following source product parameter files have been
              serviced:
VMFBLD2185R 5735FALK $PPF
VMFBLD2185R When source product parameter files are serviced, all
              product parameter files built from them must be recompiled
              using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
              parameter files built to your A-disk and exit VMFBLD so
              you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
              recompiled your product parameter files with VMFPPF.
```

**0**

Enter a 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 5735FALK $PPF
              on 191 (A) from level $PFnnnnn
```

**Note:** If you are installing corrective service for IBM Network Station support (APAR PQ01770 — initial PTF **UQ03096**), you need to create a PPF override for the 5735FALK \$PPF file that's been updated with this service. See Appendix G, "Overriding the TFTP Disk Link Address" on page 201 for more information about the need for this override file, and how to create it.

After you have created the required \$PPF override and re-compiled the necessary PPF files, you then can continue with this procedure.

**vmfppf 5735falk \***

**Note:** If you've created your own PPF override, use your PPF name instead of 5735FALK.

**copyfile 5735falk \$ppf a = = d (olddate replace  
erase 5735falk \$ppf a**

**Note:** **Do not** use your own PPF name in place of 5735FALK for the COPYFILE and ERASE commands.

Note Continued ...

**vmfbl d ppf 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} (status setup**

**1**

Re-issue VMFBLD to complete updating the build status table.

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS). When you receive the prompt that was previously displayed, enter a 1 to continue.

**2** Use VMFVIEW to review the build status messages, and see what objects need to be built.

**vmfview build**

### 7.2.3.5 Build Serviced Objects

**1** Rebuild TCP/IP V2 R4 for VM serviced parts.

**Note:** If you're applying service to functions that require Language Environment runtime library support, ensure this support is available when you perform this step.

**vmfbl d ppf 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} (serviced**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and you want to apply service that requires Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

**2** Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview build**

### 7.2.3.6 Test the New Service

You should thoroughly test the service before putting it into production. To do this, add links in each server machine that you are using to link the test build disks for the server and client code instead of the production build disks. You will have to detach the 591 and 592 disks first. The link commands to issue are:

```
LINK P735FALK 491 591 RR
LINK P735FALK 492 592 RR
```

In addition, the NCS server virtual machine should link the test NCS administration code if it was serviced. Detach the 195 disk first.

```
LINK P735FALK 395 195 RR
```

## 7.2.4 Place the New TCP/IP V2 R4 for VM Service Into Production

### 7.2.4.1 Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production

When the new service has been thoroughly tested, it should be copied to the production build disks to place it into production.

#### Note

When you use the TCP2PROD command to copy **all** files from the 491 and 492 minidisks to the 591 and 592 minidisks as part of this procedure, you will also put into production any service applied to installed TCP/IP V2 R4 for VM features (NFS, Kerberos US, and Kerberos World Trade).

- 1 Log on the TCP/IP V2 R4 for VM service user ID, **P735FALK**.
- 2 Establish read access to the VMSES/E code and the Software Inventory minidisks.

```
link maint 5e5 5e5 rr
access 5e5 b
```

The 5E5 minidisk contains the VMSES/E code.

```
link maint 51d 51d rr
access 51d d
```

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

- 3 Establish the correct minidisk access order.

```
vmfsetup 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} (link
```

Use **tcpip** or **tcpiptsfs** if you installed to minidisks or the Shared File System respectively, and have applied service to components that require Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpiptsfnoc** (SFS).

- 4 Review the 5735FALK CATALOG file to ensure any changes you made to this file remain in effect, and that any local service used to customize this file has been properly applied.

**Note:** You should verify the correctness of both the product runtime file section (TCPRUN) and the customizable sample file (TCPCONFIG) section of the 5735FALK CATALOG at the same time. For reference purposes, files processed using the TCPRUN section are listed in Figure 31 on page 59 of 6.2.5.3, “TCP/IP V2 R4 for VM Product and Sample Configuration Files” on page 58. Files processed using the TCPCONFIG section are listed in Figure 32 on page 60 of this same section.

- 5 Establish a write link to any TCP/IP for VM production or server minidisks which are not yet linked in this mode.

If you did not add the appropriate LINK statements to the P735FALK directory when TCP/IP for VM resources were allocated (based on information in the 5735FALK PLANINFO file), you will need to manually link the TCP/IP production and server minidisks necessary for your environment. See Figure 31 on page 59 and Figure 32 on page 60 for the link addresses you should use. If you have a PPF override that has changed any of these addresses, use your values.

```
| link tcpipid vdev1 vdev2 mr
```

**Note:** If another user already has a required minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the minidisk in read-only mode (RR), and then re-issue the above LINK command. Do not continue with these procedures until a R/W link is established for the minidisk in question.

- 6 Copy serviced TCP/IP for VM files into production using the TCP2PROD command. The files copied via the following command are those identified in the TCPRUN section of the 5735FALK CATALOG file.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} 5735falk tcprun**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and have applied service to components that require Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

#### **7.2.4.2 Configure TCP/IP V2 R4 for VM**

If there has been service to any of the configuration or exit files that you have customized, you need to include this service in your customized versions of these files.

See the *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for detailed explanations about how to configure the TCP/IP for VM servers.

**Note:** For step 2 below, it's assumed that the TCPCONFIG section of the 5735FALK CATALOG has been verified, as described in 4 on page 134 of 7.2.4.1, "Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production" on page 133. If this is not the case, you should make any necessary changes to the TCPCONFIG section of the 5735FALK CATALOG file before you continue with the following steps.

- 1** If necessary, establish the appropriate environment, as described by steps 1 through 3, in 7.2.4.1, "Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production" on page 133.
- 2** Copy serviced TCP/IP for VM configuration files into production using the TCP2PROD command. For reference purposes, files that can be processed using the TCPCONFIG section are listed in Figure 32 on page 60 of 6.2.5.3, "TCP/IP V2 R4 for VM Product and Sample Configuration Files" on page 58.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735falk {tcpip | tcpipsfs | tcpipnoc | tcpipsfsnoc} 5735falk tcpconfig**

Use **tcpip** or **tcpipsfs** if you installed to minidisks or the Shared File System respectively, and have applied service to components that require Language Environment runtime library support. Otherwise, use **tcpipnoc** (minidisks) or **tcpipsfsnoc** (SFS).

For information about copying client code to the Product Code minidisk, see Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182.

**You have finished servicing TCP/IP V2 R4 for VM.**

---

### **7.3 Servicing TCP/IP V2 R4 for VM - Source Feature**

The Source Feature does not have separate service. All service is provided through the service process for the TCP/IP V2 R4 for VM base product, using its PPF and components. See the 7.2, “Servicing TCP/IP V2 R4 for VM” on page 118 for a description of the service procedures for the base product.

**You have finished servicing the Source Feature.**

---

## 7.4 Servicing TCP/IP V2 R4 for VM - NFS Feature

### 7.4.1 Important Service Notes

#### 7.4.1.1 PPF Override Considerations

- The *ppfname* used throughout these servicing instructions is **5735FALL**, which assumes you are using the PPF supplied by IBM for the NFS Feature. If you have your own PPF override file for the NFS Feature, you should use your file's *ppfname* instead of **5735FALL**. The *ppfname* you use should be used **throughout** the rest of this procedure.

#### 7.4.1.2 Language Environment Runtime Library Considerations

- IBM Language Environment for MVS and VM Release 5 (5688198), or the equivalent Language Environment runtime library support, is required to service NFS Feature modules.

### 7.4.2 Preventive Service for NFS Feature (RSU)

Preventive service is available periodically on the Recommended Service Upgrade (RSU) tape. Each RSU is cumulative and contains selected important PTFs. The service on the RSU is in pre-applied, pre-built format, and includes the serviced files, the objects that were rebuilt using these files, and the updated software inventory. This makes installing the new service much quicker, but will require that you re-apply any reach-ahead service (service that you have applied that is not on the RSU). RSU tapes are in install format, so the VMFINS command is used to load the tape.

#### 7.4.2.1 Prepare to Receive Service

**1** Log on the NFS Feature service user ID, **P735FALK**.

**2** Establish access to the software inventory disk.

**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user that has it accessed R/W link it R/O. You then can issue the following commands to obtain R/W access to it.

**link maint 51d 51d mr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Have the NFS Feature RSU tape mounted and attached as 181 to **P735FALK**.

**4** Receive the product documentation (5735FALL MEMO) to the 51D minidisk and identify the products and components that have service on the tape. The product documentation contains the amount of storage needed to receive the



service on the RSU. Use this to ensure that there is enough space on the service disks or directories.

**vmfins install info (nomemo**

**nomemo** will load but not print the memos.

- 5** Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

**vmfmrchk 5735fall {vmnfs | vmnfssfs} apply (setup**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

This command copies the alternate APPLY disk to the production APPLY disk and then clears the alternate APPLY disk.

- 6** Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview mrd**

- 7** Invoke the VMFPSU command to obtain additional information about the service contained on the RSU and how it will affect your local modifications. This command creates an output file, *appid* **PSUPLAN**, which you can review. See the *VM/ESA Service Guide* for an explanation of this file.

**vmfpsu 5735fall {vmnfs | vmnfssfs}**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

This command produces an output file comparing the service on the RSU to the service on your system. The file name is *appid* **PSUPLAN**, where *appid* is specified in the PPF file.

#### 7.4.2.2 Receive the Service

- 1** Receive the service on the RSU.

Since the RSU contains pre-applied, pre-built service in install format, the VMFINS command will load the new service to the DELTA disk, the updated apply service inventory to the APPLY disk, and the pre-built objects to the appropriate test build disks.

## vmfins install ppf 5735fall {vmnfs | vmnfssfs} (nomemo nolink)

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

The NOLINK option indicates that you don't want VMFINS to link to the appropriate minidisks, only access them if not accessed.

```
VMFINS2760I VMFINS processing started
VMFREQ2805I Product 5735FALL component VMNFS passed requisite checking
Do you want to create an override for 5735FALL VMNFS (prodid 5735FALL)?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
.
.
.
VMFINS2760I VMFINS processing completed successfully
```

- 2 Review the install message log (\$VMFINS \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview install

### 7.4.2.3 Apply the Service

Since the service on the RSU is already pre-applied, this step will just reapply any reach-ahead service (service that is on your system but not on the RSU).

- 1 Reapply reach-ahead service.

## vmfapply ppf 5735fall {vmnfs | vmnfssfs}

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

This command reapplies the reach-ahead service on your system. The version vector table (VVT) is updated with all serviced parts and all necessary AUX files are generated on the alternate apply disk.

- 2 Review the apply message log (\$VMFAPP \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

## vmfview apply

- 3** Re-work and re-apply local service, if it has been affected by the service on the RSU.
  - a. Affected local modifications are indicated in the output file created by the VMFPSU command invoked in a step 7 on page 139.
  - b. For information on re-working local modifications, refer to Chapter 7 in the *VM/ESA Service Guide*. Follow the steps that are applicable to your local modification.

The following substitutions may need to be made:

- **esalcl** should be **5735fall**
- **esa** should be **5735fall**
- *compname* should be **vmnfs** or **vmnfssfs**
- *appid* should be **5735fall**
- *fm-local* should be the fm of 2C4
- *fm-applyalt* should be the fm of 4A6

Keep in mind that when you get to the following step in the *VM/ESA Service Guide*:

- "Rebuilding Objects" (VM/ESA Version 2 Release 1.0, or later)
- "Return to the Appropriate Section to Build Remaining Objects" (VM/ESA Version 1 Release 2.2)

you should return back to this program directory and continue with 7.4.3, "Update the Build Status Table."

### 7.4.3 Update the Build Status Table

- 1** Update the Build Status Table with serviced parts.

**vmfbld ppf 5735fall {vmnfs | vmnfssfs} (status**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

This command updates the Build Status Table to determine what remains to be built.

**Note**

If a \$PPF file has been serviced you will get the following prompt:

```
VMFBLD2185R The following source product parameter files have been
serviced:
VMFBLD2185R 5735FALL $PPF
VMFBLD2185R When source product parameter files are serviced, all
product parameter files built from them must be recompiled
using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
parameter files built to your A-disk and exit VMFBLD so
you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
recompiled your product parameter files with VMFPPF.
```

**0**

Enter a 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 5735FALL $PPF
on 191 (A) from level $PFnnnnn
```

**vmfppf 5735fall \***

**Note:** If you've created your own PPF override, use your PPF name instead of 5735FALL.

Note Continued ...

**copyfile 5735fall \$ppf a = = d (olddate replace  
erase 5735fall \$ppf a**

**Note:** Do not use your own PPF name in place of 5735FALL for the COPYFILE and ERASE commands.

**vmfbld ppf 5735fall {vmnfs | vmnfssfs} (status**

**1**

Re-issue VMFBLD to complete updating the build status table.

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

When you receive the prompt that was previously displayed, enter a 1 to continue.

- 2** Use VMFVIEW to review the build status messages, and see what objects need to be built.

**vmfview build**

### 7.4.3.1 Build Serviced Objects

The RSU contains pre-built service, so this step will build only objects that were affected by any reach-ahead and local service that was reapplied.

- 1** Rebuild NFS Feature serviced parts.

**vmfbld ppf 5735fall {vmnfs | vmnfssfs} (serviced**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

- 2** Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview build**

### 7.4.3.2 Test the New Service

You should thoroughly test the service before putting it into production. To do this, add links in the VMNFS server machine to link the test build disks for the server and client code instead of the production build disks. You will have to detach the 591 and 592 disks first. The link commands to issue are:

```
LINK P735FALK 491 591 RR
LINK P735FALK 492 592 RR
```

### 7.4.3.3 Place the Service into Production

When you have thoroughly tested the new service, you should put the service into production. See 7.4.6, "Place the New NFS Feature Service Into Production" on page 150 for an explanation of this task.

### 7.4.3.4 Determine RSU Service Level

The service contained on each RSU constitutes a new service level. Use this service level when ordering corrective service. The service level is updated in the system inventory when the RSU is installed.

The following command is used to query the current service level of the system.

```
vmfsim query vm sysrecs tdata :ppf 5735fall :stat
```

```
VMFSIP2408I RESULTS FOR
      TDATA :PPF 5735FALL :STAT
:PPF 5735FALL VMNFS
:STAT RECEIVED.06/28/96.10:34:12.P735FALK.210-9602
```

The last part of the status line indicates the RSU service level: 9602.

## 7.4.4 Corrective Service for NFS Feature

Corrective service for NFS Feature is provided in COR format via tape or electronic envelope. It is installed using VMFREC, VMFAPPLY, and VMFBLD.

### 7.4.4.1 Prepare to Receive Service

**1** Log on the NFS Feature service user ID, **P735FALK**.

**2** Establish access to the software inventory disk.

**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user that has it accessed R/W link it R/O. You then can issue the following commands to obtain R/W access to it.

**link maint 51d 51d mr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Have the NFS Feature CORrective service tape mounted and attached as 181 to **P735FALK**.

**4** Establish the correct minidisk access order.

**vmfsetup 5735fall {vmnfs | vmnfssfs}**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

**5** Receive the documentation. VMFREC, with the INFO option, loads the documentation and displays a list of all the products on the tape.

**Electronic Service**

If you are receiving service from ServiceLink (electronic service) see Appendix A, "Receiving Service for VMSES Envelopes," section Receive Service Documentation, in the *VM/ESA Service Guide*. Then continue with step 7.

**vmfrec info**

This command will load the service memo to the 191 disk.

**6** Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview receive**

Also make note of which products and components have service on the tape. To do this, use the PF5 key to show all status messages which identify the products on the tape.

**7** Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

**vmfmrsk 5735fall {vmnfs | vmnfssfs} apply**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

This command copies the alternate APPLY disk to the production APPLY disk and then clears the alternate APPLY disk.

- 8 Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview mrd**

#### 7.4.4.2 Receive the Service

##### Electronic Service

If you are receiving service from ServiceLink (electronic service) see Appendix A, "Receiving Service for VMSES Envelopes," in the *VM/ESA Service Guide*. Then continue with 7.4.4.3, "Apply the Service" on page 147.

- 1 Receive the service.

**vmfrec ppf 5735fall {vmnfs | vmnfssfs}**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

This command receives service from your service tape. All new service is loaded to the DELTA disk.

- 2 Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview receive**



### 7.4.4.3 Apply the Service

**1** Apply the new service.

**vmfapply ppf 5735fall {vmnfs | vmnfssfs}**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

This command applies the service that you just received. The version vector table (VVT) is updated with all serviced parts and all necessary AUX files are generated on the alternate apply disk.

**2** Review the apply message log (`$VMFAPP $MSGLOG`). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview apply**

#### Note

If you receive message VMFAPP2120W, you need to re-apply any local modifications before building the new NFS Feature. Refer to chapter 7 in the *VM/ESA Service Guide*. Follow the steps that are applicable to your local modification.

The following substitutions may need to be made:

- **esalcl** should be **5735fall**
- **esa** should be **5735fall**
- *compname* should be **vmnfs** or **vmnfssfs**
- *appid* should be **5735fall**
- *fm-local* should be the fm of 2C4
- *fm-applyalt* should be the fm of 4A6

Keep in mind that when you get to the following step in the *VM/ESA Service Guide*:

- "Rebuilding Objects" (VM/ESA Version 2 Release 1.0, or later)
- "Return to the Appropriate Section to Build Remaining Objects" (VM/ESA Version 1 Release 2.2)

you should return back to this program directory and continue with 7.4.5, "Update the Build Status Table."

### 7.4.5 Update the Build Status Table

**1** Update the Build Status Table with serviced parts.

**vmfbld ppf 5735fall {vmnfs | vmnfssfs} (status**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

This command updates the Build Status Table to include objects that have to be built due to the new service just applied.

**Note**

If a \$PPF file has been serviced you will get the following prompt:

```
VMFBLD2185R The following source product parameter files have been
              serviced:
VMFBLD2185R 5735FALL $PPF
VMFBLD2185R When source product parameter files are serviced, all
              product parameter files built from them must be recompiled
              using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
              parameter files built to your A-disk and exit VMFBLD so
              you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
              recompiled your product parameter files with VMFPPF.
```

**0**

Enter a 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 5735FALL $PPF
              on 191 (A) from level $PFnnnnn
```

**vmfppf 5735fall \***

**Note:** If you've created your own PPF override, use your PPF name instead of 5735FALL.

**copyfile 5735fall \$ppf a = d (olddate replace  
erase 5735fall \$ppf a**

**Note:** Do not use your own PPF name in place of 5735FALL for the COPYFILE and ERASE commands.

**vmfbld ppf 5735fall {vmnfs | vmnfssfs} (status**

**1**

Re-issue VMFBLD to complete updating the build status table.

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

When you receive the prompt that was previously displayed, enter a 1 to continue.

- 2 Use VMFVIEW to review the build status messages, and see what objects need to be built.

**vmfview build**

### 7.4.5.1 Build Serviced Objects

**Note:** Ensure Language Environment runtime library support is available when you perform this step.

- 1 Rebuild NFS Feature serviced parts.

**vmfbld ppf 5735fall {vmnfs | vmnfssfs}  
(serviced**

Use **vmnfs** or **vmnfssfs** if you installed to minidisks or the Shared File System respectively.

- 2 Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview build**

### 7.4.5.2 Test the New Service

You should thoroughly test the service before putting it into production. To do this, add links in the VMNFS server machine to link the test build disks for the server and client code instead of the production build disks. You will have to detach the 591 and 592 disks first. The link commands to issue are:

```
LINK P735FALK 491 591 RR  
LINK P735FALK 492 592 RR
```

## 7.4.6 Place the New NFS Feature Service Into Production

### 7.4.6.1 Copy the New NFS Feature Serviced Files Into Production

When the new service has been thoroughly tested, it should be copied to the production build disks to place it into production.

- 1 Log on the NFS Feature service user ID, **P735FALK**.
- 2 Establish read access to the VMSES/E code and the Software Inventory minidisks.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

**link maint 51d 51d rr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Establish the correct minidisk access order.

**vmfsetup 5735fall {vmnfs | vmnfssfs} (link**

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmnfssfs** if you installed it to Shared File System directories.

**4** Review the 5735FALL CATALOG file to ensure any changes you made to this file remain in effect, and that any local service used to customize this file has been properly applied.

**Note:** You should verify the correctness of both the product runtime file section (NFSRUN) and the customizable sample file (NFSCONFIG) section of the 5735FALL CATALOG at the same time. For reference purposes, files processed using the NFSRUN section are listed in Figure 33 on page 82 of 6.4.4.3, "NFS Feature Runtime and Sample Configuration Files" on page 82. Files processed using the NFSCONFIG section are listed in Figure 34 on page 82 of this same section.

**5** Establish a write link to any TCP/IP for VM production or server minidisks which are not yet linked in this mode.

If you did not add the appropriate LINK statements to the P735FALK directory when TCP/IP for VM resources were allocated (based on information in the 5735FALL PLANINFO file), you will need to manually link the TCP/IP production and server minidisks necessary for your environment. See Figure 33 on page 82 and Figure 34 on page 82 for the link addresses you should use. If you have a PPF override that has changed any of these addresses, use your values.

**link tcpipid vdev1 vdev2 mr**

**Note:** If another user already has a required minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the minidisk in read-only mode (RR), and then re-issue the above LINK command. Do not continue with these procedures until a R/W link is established for the minidisk in question.

- 6 Copy serviced TCP/IP for VM files into production using the TCP2PROD command. The files copied via the following command are those identified in the NFSRUN section of the 5735FALL CATALOG file.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

`tcp2prod 5735fall {vmnfs | vmnfssfs} 5735fall nfsrun`

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmnfssfs** if you installed it to Shared File System directories.

#### 7.4.6.2 Configure the NFS Feature

If there has been service to any of the configuration or exit files that you have customized, you need to include this service in your customized versions of these files.

See the *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for a detailed explanation about how to configure the VMNFS server.

**Note:** For step 2 below, it's assumed that the NFSCONFIG section of the 5735FALL CATALOG has been verified, as described in 4 on page 151 of 7.2.4.1, "Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production" on page 133. If this is not the case, you should make any necessary changes to the NFSCONFIG section of the 5735FALL CATALOG file before you continue with the following steps.

- 1 If necessary, establish the appropriate environment, as described by steps 1 through 3, in 7.2.4.1, "Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production" on page 133.
- 2 Copy serviced NFS Feature configuration files into production using the TCP2PROD command. For reference purposes, files that can be processed using the NFSCONFIG section are listed in Figure 34 on page 82 of 6.4.4.3, "NFS Feature Runtime and Sample Configuration Files" on page 82.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735fall {vmnfs | vmnfssfs} 5735fall nfsconfig**

Use **vmnfs** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmnfssfs** if you installed it to Shared File System directories.

For information about copying client code to the Product Code minidisk, see Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182.

**You have finished servicing the NFS Feature.**

---

## 7.5 Servicing TCP/IP V2 R4 for VM - Kerberos US Feature

### 7.5.1 Important Service Notes

- Service for the Kerberos US Feature is provided only in Corrective service (COR) format.

#### 7.5.1.1 PPF Override Considerations

- The *ppfname* used throughout these servicing instructions is **5735FALN**, which assumes you are using the PPF supplied by IBM for the Kerberos US Feature. If you have your own PPF override file for the Kerberos US Feature, you should use your file's *ppfname* instead of **5735FALN**. The *ppfname* you use should be used **throughout** the rest of this procedure.

#### 7.5.1.2 Language Environment Runtime Library Considerations

- IBM Language Environment for MVS and VM Release 5 (5688198), or the equivalent Language Environment runtime library support, is required to service Kerberos US Feature modules.

### 7.5.2 Corrective Service for Kerberos US Feature

Corrective service for Kerberos US Feature is provided in COR format via tape or electronic envelope. It is installed using VMFREC, VMFAPPLY, and VMFBLD.

#### 7.5.2.1 Prepare to Receive Service

**1** Log on the Kerberos US Feature service user ID, **P735FALK**.

**2** Establish access to the software inventory disk.

**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user that has it accessed R/W link it R/O. You then can issue the following commands to obtain R/W access to it.

```
link maint 51d 51d mr  
access 51d d
```

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Have the Kerberos US Feature CORrective service tape mounted and attached as 181 to **P735FALK**.

**4** Establish the correct minidisk access order.



**vmfsetup 5735faln {vmkerb | vmkerbsfs}**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

- 5 Receive the documentation. VMFREC, with the INFO option, loads the documentation and displays a list of all the products on the tape.

**Electronic Service**

If you are receiving service from ServiceLink (electronic service) see Appendix A, "Receiving Service for VMSES Envelopes," section Receive Service Documentation, in the *VM/ESA Service Guide*. Then continue with step 7.

**vmfrec info**

This command will load the service memo to the 191 disk.

- 6 Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview receive**

Also make note of which products and components have service on the tape. To do this, use the PF5 key to show all status messages which identify the products on the tape.

- 7 Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

**vmfmrdsd 5735faln {vmkerb | vmkerbsfs} apply**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

This command copies the alternate APPLY disk to the production APPLY disk and then clears the alternate APPLY disk.

- 8 Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview mrd**

## 7.5.2.2 Receive the Service

### Electronic Service

If you are receiving service from ServiceLink (electronic service) see Appendix A, "Receiving Service for VMSES Envelopes," in the *VM/ESA Service Guide*. Then continue with 7.5.2.3, "Apply the Service."

- 1 Receive the service.

**vmfrec ppf 5735faln {vmkerb | vmkerbsfs}**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

This command receives service from your service tape. All new service is loaded to the DELTA disk.

- 2 Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview receive**

## 7.5.2.3 Apply the Service

- 1 Apply the new service.

**vmfapply ppf 5735faln {vmkerb | vmkerbsfs}**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

This command applies the service that you just received. The version vector table (VVT) is updated with all serviced parts and all necessary AUX files are generated on the alternate apply disk.

- 2 Review the apply message log (\$VMFAPP \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview apply**

#### Note

If you receive message VMFAPP2120W, you need to re-apply any local modifications before building the new Kerberos US Feature. Refer to chapter 7 in the *VM/ESA Service Guide*. Follow the steps that are applicable to your local modification.

The following substitutions may need to be made:

- **esalcl** should be **5735faln**
- **esa** should be **5735faln**
- *compname* should be **vmkerb** or **vmkerbsfs**
- *appid* should be **5735faln**
- *fm-local* should be the fm of 2C4
- *fm-applyalt* should be the fm of 4A6

Keep in mind that when you get to the following step in the *VM/ESA Service Guide*:

- "Rebuilding Objects" (VM/ESA Version 2 Release 1.0, or later)
- "Return to the Appropriate Section to Build Remaining Objects" (VM/ESA Version 1 Release 2.2)

you should return back to this program directory and continue with 7.5.3, "Update the Build Status Table" on page 157.

### 7.5.3 Update the Build Status Table

- 1 Update the Build Status Table with serviced parts.

**vmfbld ppf 5735faln {vmkerb | vmkerbsfs}**  
**(status**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

This command updates the Build Status Table to include objects that have to be built due to the new service just applied.

**Note**

If a \$PPF file has been serviced you will get the following prompt:

```
VMFBLD2185R The following source product parameter files have been
              serviced:
VMFBLD2185R 5735FALN $PPF
VMFBLD2185R When source product parameter files are serviced, all
              product parameter files built from them must be recompiled
              using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
              parameter files built to your A-disk and exit VMFBLD so
              you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
              recompiled your product parameter files with VMFPPF.
```

**0**

Enter a 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 5735FALN $PPF
              on 191 (A) from level $PFnnnnn
```

**vmfppf 5735faln \***

**Note:** If you've created your own PPF override, use your PPF name instead of 5735FALN.

**copyfile 5735faln \$ppf a = = d (olddate replace  
erase 5735faln \$ppf a**

**Note:** Do not use your own PPF name in place of 5735FALN for the COPYFILE and ERASE commands.

**vmfbld ppf 5735faln {vmkerb | vmkerbsfs} (status**

**1**

Re-issue VMFBLD to complete updating the build status table.

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

When you receive the prompt that was previously displayed, enter a 1 to continue.

- 2 Use VMFVIEW to review the build status messages, and see what objects need to be built.

**vmfview build**

### 7.5.3.1 Build Serviced Objects

**Note:** Ensure Language Environment runtime library support is available when you perform this step.

- 1 Rebuild Kerberos US Feature serviced parts.

**vmfblid ppf 5735faln {vmkerb | vmkerbsfs} (serviced**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

- 2 Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview build**

### 7.5.3.2 Test the New Service

You should thoroughly test the service before putting it into production. To do this, add links in the VMKERB and ADMSERV server machines to link the test build disks for the server and client code instead of the production build disks. You will have to detach the 591 and 592 disks first. The link commands to issue are:

```
LINK P735FALK 491 591 RR  
LINK P735FALK 492 592 RR
```

## 7.5.4 Place the New Kerberos US Feature Service Into Production

### 7.5.4.1 Copy the New Kerberos US Feature Serviced Files Into Production

When the new service has been thoroughly tested, it should be copied to the production build disks to place it into production.

- 1 Log on the Kerberos US Feature service user ID, **P735FALK**.
- 2 Establish read access to the VMSES/E code and the Software Inventory minidisks.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

**link maint 51d 51d rr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Establish the correct minidisk access order.

**vmfsetup 5735faln {vmkerb | vmkerbsfs} (link**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

**4** Review the 5735FALN CATALOG file to ensure any changes you made to this file remain in effect, and that any local service used to customize this file has been properly applied.

**Note:** You should verify the correctness of both the product runtime file section (KERBRUN) and the customizable sample file (KERBCONFIG) section of the 5735FALN CATALOG at the same time. For reference purposes, files processed using the KERBRUN section are listed in Figure 35 on page 97 of 6.5.4.3, "Kerberos US Feature Runtime and Sample Configuration Files" on page 97. Files processed using the KERBCONFIG section are listed in Figure 36 on page 98 of this same section.

**5** Establish a write link to any TCP/IP for VM production or server minidisks which are not yet linked in this mode.

If you did not add the appropriate LINK statements to the P735FALK directory when TCP/IP for VM resources were allocated (based on information in the 5735FALN PLANINFO file), you will need to manually link the TCP/IP production and server minidisks necessary for your environment. See Figure 35 on page 97 and Figure 36 on page 98 for the link addresses you should use. If you have a PPF override that has changed any of these addresses, use your values.

**link tcpipid vdev1 vdev2 mr**

**Note:** If another user already has a required minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the minidisk in read-only mode (RR), and then re-issue the above LINK command. Do not continue with these procedures until a R/W link is established for the minidisk in question.

- 6 Copy serviced TCP/IP for VM files into production using the TCP2PROD command. The files copied via the following command are those identified in the KERBRUN section of the 5735FALN CATALOG file.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735faln {vmkerb | vmkerbsfs} 5735faln kerbrun**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

#### 7.5.4.2 Configure the Kerberos US Feature

If there has been service to any of the configuration or exit files that you have customized, you need to include this service in your customized versions of these files.

See the *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for detailed explanations about how to configure the Kerberos servers.

**Note:** For step 2 below, it's assumed that the KERBCONFIG section of the 5735FALN CATALOG has been verified, as described in 4 on page 160 of 7.2.4.1, "Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production" on page 133. If this is not the case, you should make any necessary changes to the KERBCONFIG section of the 5735FALN CATALOG file before you continue with the following steps.

- 1 If necessary, establish the appropriate environment, as described by steps 1 through 3, in 7.2.4.1, "Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production" on page 133.
- 2 Copy serviced Kerberos US Feature configuration files into production using the TCP2PROD command. For reference purposes, files that can be processed using the KERBCONFIG section are listed in Figure 36 on page 98 of 6.5.4.3, "Kerberos US Feature Runtime and Sample Configuration Files" on page 97.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735faln {vmkerb | vmkerbsfs} 5735faln kerbconfig**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

For information about copying client code to the Product Code minidisk, see Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182.

**You have finished servicing the Kerberos US Feature.**



---

## 7.6 Servicing TCP/IP V2 R4 for VM - Kerberos WT Feature

### 7.6.1 Important Service Notes

- Service for the Kerberos WT Feature is provided only in Corrective service (COR) format.

#### 7.6.1.1 PPF Override Considerations

- The *ppfname* used throughout these servicing instructions is **5735FALP**, which assumes you are using the PPF supplied by IBM for the Kerberos WT Feature. If you have your own PPF override file for the Kerberos WT Feature, you should use your file's *ppfname* instead of **5735FALP**. The *ppfname* you use should be used **throughout** the rest of this procedure.

#### 7.6.1.2 Language Environment Runtime Library Considerations

- IBM Language Environment for MVS and VM Release 5 (5688198), or the equivalent Language Environment runtime library support, is required to service Kerberos WT Feature modules.

### 7.6.2 Corrective Service for Kerberos WT Feature

Corrective service for Kerberos WT Feature is provided in COR format via tape or electronic envelope. It is installed using VMFREC, VMFAPPLY, and VMFBLD.

#### 7.6.2.1 Prepare to Receive Service

**1** Log on the Kerberos WT Feature service user ID, **P735FALK**.

**2** Establish access to the software inventory disk.

**Note:** If the MAINT 51D minidisk was accessed R/O, you will need to have the user that has it accessed R/W link it R/O. You then can issue the following commands to obtain R/W access to it.

```
link maint 51d 51d mr  
access 51d d
```

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Have the Kerberos WT Feature CORrective service tape mounted and attached as 181 to **P735FALK**.

**4** Establish the correct minidisk access order.

**vmfsetup 5735falp {vmkerb | vmkerbsfs}**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

- 5 Receive the documentation. VMFREC, with the INFO option, loads the documentation and displays a list of all the products on the tape.

**Electronic Service**

If you are receiving service from ServiceLink (electronic service) see Appendix A, "Receiving Service for VMSES Envelopes," section Receive Service Documentation, in the *VM/ESA Service Guide*. Then continue with step 7.

**vmfrec info**

This command will load the service memo to the 191 disk.

- 6 Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview receive**

Also make note of which products and components have service on the tape. To do this, use the PF5 key to show all status messages which identify the products on the tape.

- 7 Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

**vmfmrdsk 5735falp {vmkerb | vmkerbsfs} apply**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

This command copies the alternate APPLY disk to the production APPLY disk and then clears the alternate APPLY disk.

- 8 Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview mrd**

## 7.6.2.2 Receive the Service

### Electronic Service

If you are receiving service from ServiceLink (electronic service) see Appendix A, "Receiving Service for VMSES Envelopes," in the *VM/ESA Service Guide*. Then continue with 7.6.2.3, "Apply the Service."

- 1 Receive the service.

**vmfrec ppf 5735falp {vmkerb | vmkerbsfs}**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

This command receives service from your service tape. All new service is loaded to the DELTA disk.

- 2 Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview receive**

## 7.6.2.3 Apply the Service

- 1 Apply the new service.

**vmfapply ppf 5735falp {vmkerb | vmkerbsfs}**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

This command applies the service that you just received. The version vector table (VVT) is updated with all serviced parts and all necessary AUX files are generated on the alternate apply disk.

- 2 Review the apply message log (\$VMFAPP \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview apply**

### Note

If you receive message VMFAPP2120W, you need to re-apply any local modifications before building the new Kerberos WT Feature. Refer to chapter 7 in the *VM/ESA Service Guide*. Follow the steps that are applicable to your local modification.

The following substitutions may need to be made:

- **esalcl** should be **5735falp**
- **esa** should be **5735falp**
- *compname* should be **vmkerb** or **vmkerbsfs**
- *appid* should be **5735falp**
- *fm-local* should be the fm of 2C4
- *fm-applyalt* should be the fm of 4A6

Keep in mind that when you get to the following step in the *VM/ESA Service Guide*:

- "Rebuilding Objects" (VM/ESA Version 2 Release 1.0, or later)
- "Return to the Appropriate Section to Build Remaining Objects" (VM/ESA Version 1 Release 2.2)

you should return back to this program directory and continue with 7.6.2.4, "Update the Build Status Table" on page 166.

## 7.6.2.4 Update the Build Status Table

### 1 Update the Build Status Table with serviced parts.

**vmfbld ppf 5735falp {vmkerb | vmkerbsfs}**  
**(status**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

This command updates the Build Status Table to include objects that have to be built due to the new service just applied.

## Note

If a \$PPF file has been serviced you will get the following prompt:

```
VMFBLD2185R The following source product parameter files have been
              serviced:
VMFBLD2185R 5735FALP $PPF
VMFBLD2185R When source product parameter files are serviced, all
              product parameter files built from them must be recompiled
              using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
              parameter files built to your A-disk and exit VMFBLD so
              you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
              recompiled your product parameter files with VMFPPF.
```

**0**

Enter a 0 and complete the following steps before you continue.

```
VMFBLD2188I Building 5735FALP $PPF
              on 191 (A) from level $PFnnnnn
```

**vmfppf 5735falp \***

**Note:** If you've created your own PPF override, use your PPF name instead of 5735FALP.

**copyfile 5735falp \$ppf a = = d (olddate replace  
erase 5735falp \$ppf a**

**Note:** Do not use your own PPF name in place of 5735FALP for the COPYFILE and ERASE commands.

**vmfbld ppf 5735falp {vmkerb | vmkerbsfs} (status**

**1**

Re-issue VMFBLD to complete updating the build status table.

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

When you receive the prompt that was previously displayed, enter a 1 to continue.

- 2 Use VMFVIEW to review the build status messages, and see what objects need to be built.

**vmfview build**

### 7.6.2.5 Build Serviced Objects

**Note:** Ensure Language Environment runtime library support is available when you perform this step.

- 1 Rebuild Kerberos WT Feature serviced parts.

**vmfblid ppf 5735falp {vmkerb | vmkerbsfs} (serviced**

Use **vmkerb** or **vmkerbsfs** if you installed to minidisks or the Shared File System respectively.

- 2 Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA System Messages and Codes* or use on-line HELP.

**vmfview build**

### 7.6.2.6 Test the New Service

You should thoroughly test the service before putting it into production. To do this, add links in the VMKERB and ADMSERV server machines to link the test build disks for the server and client code instead of the production build disks. You will have to detach the 591 and 592 disks first. The link commands to issue are:

```
LINK P735FALK 491 591 RR  
LINK P735FALK 492 592 RR
```

## 7.6.3 Place the New Kerberos WT Feature Service Into Production

### 7.6.3.1 Copy the New Kerberos WT Feature Serviced Files Into Production

When the new service has been thoroughly tested, it should be copied to the production build disks to place it into production.

- 1 Log on the Kerberos WT Feature service user ID, **P735FALK**.
- 2 Establish read access to the VMSES/E code and the Software Inventory minidisks.

**link maint 5e5 5e5 rr  
access 5e5 b**

The 5E5 minidisk contains the VMSES/E code.

**link maint 51d 51d rr  
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

### **3** Establish the correct minidisk access order.

**vmfsetup 5735falp {vmkerb | vmkerbsfs} (link**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

### **4** Review the 5735FALP CATALOG file to ensure any changes you made to this file remain in effect, and that any local service used to customize this file has been properly applied.

**Note:** You should verify the correctness of both the product runtime file section (KERBRUN) and the customizable sample file (KERBCONFIG) section of the 5735FALP CATALOG at the same time. For reference purposes, files processed using the KERBRUN section are listed in Figure 37 on page 113 of 6.6.4.3, “Kerberos WT Feature Runtime and Sample Configuration Files” on page 113. Files processed using the KERBCONFIG section are listed in Figure 38 on page 114 of this same section.

### **5** Establish a write link to any TCP/IP for VM production or server minidisks which are not yet linked in this mode.

If you did not add the appropriate LINK statements to the P735FALK directory when TCP/IP for VM resources were allocated (based on information in the 5735FALP PLANINFO file), you will need to manually link the TCP/IP production and server minidisks necessary for your environment. See Figure 37 on page 113 and Figure 38 on page 114 for the link addresses you should use. If you have a PPF override that has changed any of these addresses, use your values.

**link tcpipid vdev1 vdev2 mr**

**Note:** If another user already has a required minidisk linked in write mode (R/W), you'll only obtain read access (R/O) to this minidisk. If this occurs, you'll need to have that user re-link the minidisk in read-only mode (RR), and then re-issue the above LINK command. Do not continue with these procedures until a R/W link is established for the minidisk in question.

- 6 Copy serviced TCP/IP for VM files into production using the TCP2PROD command. The files copied via the following command are those identified in the KERBRUN section of the 5735FALP CATALOG file.

**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735falp {vmkerb | vmkerbsfs} 5735falp kerbrun**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

### 7.6.3.2 Configure the Kerberos WT Feature

If there has been service to any of the configuration or exit files that you have customized, you need to include this service in your customized versions of these files.

See the *TCP/IP V2 R4 for VM: Planning and Customization* (SC31-6082) for detailed explanations about how to configure the Kerberos servers.

**Note:** For step 2 below, it's assumed that the KERBCONFIG section of the 5735FALP CATALOG has been verified, as described in 4 on page 169 of 7.2.4.1, "Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production" on page 133. If this is not the case, you should make any necessary changes to the KERBCONFIG section of the 5735FALP CATALOG file before you continue with the following steps.

- 1 If necessary, establish the appropriate environment, as described by steps 1 through 3, in 7.2.4.1, "Copy the New TCP/IP V2 R4 for VM Serviced Files Into Production" on page 133.
- 2 Copy serviced Kerberos WT Feature configuration files into production using the TCP2PROD command. For reference purposes, files that can be processed using the KERBCONFIG section are listed in Figure 38 on page 114 of 6.6.4.3, "Kerberos WT Feature Runtime and Sample Configuration Files" on page 113.



**Note**

When you perform this step, you should first issue the TCP2PROD command with the **TEST** option, to verify that all resources can be accessed and that the appropriate files will be processed. With the **TEST** option in effect, **no files are copied into production**. Resolve any reported problems, then invoke TCP2PROD without the TEST option, as illustrated below.

**tcp2prod 5735falp {vmkerb | vmkerbsfs} 5735falp kerbconfig**

Use **vmkerb** if you installed the TCP/IP V2 R4 for VM base product on minidisks; use **vmkerbsfs** if you installed it to Shared File System directories.

For information about copying client code to the Product Code minidisk, see Appendix C, "Copying TCP/IP for VM Client Code to the Y-Disk" on page 182.

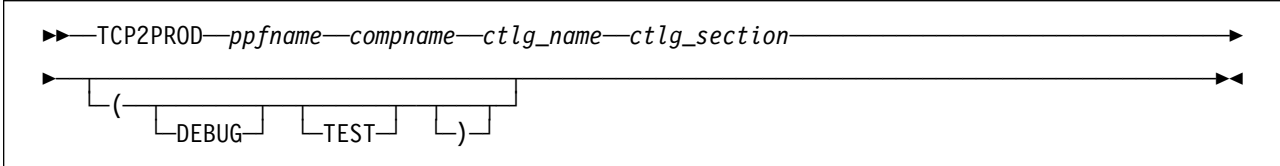
**You have finished servicing the Kerberos WT Feature.**

---

## Appendix A. TCP2PROD

---

### A.1 TCP2PROD



#### A.1.1 Purpose

Use the TCP2PROD command to help you place TCP/IP Version 2 Release 4 for VM files into production. The TCP2PROD command uses the content of a TCP/IP for VM CATALOG file to determine which product files are to be placed into production, as well as which minidisks and SFS directories are to be used.

**Note:** The TCP2PROD command is intended for use by the P735FALK user ID, and should only be used when you install or service TCP/IP for VM and its features.

#### A.1.2 Operands

*ppfname* is the name of the usable form product parameter file to be used for this purpose. The file type must be PPF.

*compname* is the name of the component as specified on the :COMPNAME tag in the product parameter file. *compname* is a 1- to 16-character alphanumeric identifier.

The PPF Variable Declarations (:DCL.) section defined for *compname* determines from which source minidisks and SFS directories product files are copied; likewise for the target minidisks to which these files are copied.

*ctlg\_name* is the name of the product catalog file to be processed. The file type must be CATALOG.

*ctlg\_section* is the section of the catalog file to be processed. The value specified by *ctlg\_section* is used as a “root” for the “begin” and “end” tags that define each section of grouped records within a catalog file.

### A.1.3 Options

#### *TEST*

causes processing to be performed so that no files are placed into production via the VMSES/E VMFCOPY command. This option allows you to verify that minidisks and SFS directories can be accessed as required, and that the appropriate catalog file records will be processed. Additional messages are issued in some cases to clarify what processing would be performed if the TEST option was not used.

#### *DEBUG*

causes additional messages to be issued to provide information for debugging purposes.

### A.1.4 Usage Notes

- **TCP2PROD does not issue LINK commands** as part of its processing.

If necessary, the appropriate LINK statements should be added to the P735FALK directory entry. Alternatively, the required LINK commands can be added to the P735FALK user ID's PROFILE EXEC.

- You should issue a VMFSETUP command prior to invoking TCP2PROD, so that the appropriate environment is established for placing files into production.
- The first CATALOG file found in the CMS search order that matches the name specified by *ctlg\_name* is used.
- The value specified for *ctlg\_name* is also used by TCP2PROD as the *prodid* parameter when VMFCOPY commands are issued. If *ctlg\_name* is not recognized as a TCP/IP for VM catalog file name, the base product ID (5735FALK) is used for VMFCOPY commands.
- CATALOG file records which are found to be unusable are bypassed. Warning messages are issued when records are bypassed for this reason.
- The minidisk or directory accessed as file mode A must be read/write.
- If TCP2PROD is invoked with *ppfname* specified as a question mark (?), the command syntax is displayed.

### A.1.5 The TCP2PROD \$MSGLOG

Informational, warning and error messages issued to the console by TCP2PROD are maintained in a message log, TCP2PROD \$MSGLOG. This message log is written to the minidisk or directory accessed as file mode A.

The TCP2PROD \$MSGLOG is cumulative. The most recent entries are appended at the **bottom** of the file. Separator lines and date and time stamps are inserted in the log with each TCP2PROD invocation so that earlier log entries can be distinguished from new entries.

## TCP2PROD

No messages are logged until initial validation of the TCP2PROD command is complete.

### Note

The TCP2PROD \$MSGLOG does not have the same format as the VMSES/E \$VMFxxx \$MSGLOG files, so the VMFVIEW EXEC should not be used to browse this file. Use XEDIT to view the content of the TCP2PROD \$MSGLOG.

### A.1.6 Return Codes

The possible return codes produced by TCP2PROD are indicated below.

---

**0**

**Explanation:** Successful execution. No processing errors were encountered.

---

**1**

**Explanation:** Incorrect invocation. TCP2PROD was invoked with an incorrect number of parameters. A message indicating the missing parameter is displayed, in addition to the command syntax.

---

**2**

**Explanation:** Internal error. If return code 2 is returned, processing status is unknown. You should contact the TCP/IP for VM Support Group for problem determination and assistance.

---

**4**

**Explanation:** Errors encountered with warnings issued. The errors encountered may have caused processing to complete with only partial success. Review the TCP2PROD \$MSGLOG for warning messages which indicate any problems that were encountered.

---

**8**

**Explanation:** Errors encountered. Processing has not completed successfully. Review the TCP2PROD \$MSGLOG for messages regarding the problems encountered.

## A.2 TCP/IP for VM CATALOG Files

### A.2.1 Purpose

The TCP/IP for VM CATALOG files are used by TCP2PROD to determine which TCP/IP for VM product files are to be placed into production, as well as which minidisks and SFS directories are to be used for this process.

The CATALOG files provided for use with TCP/IP V2 R4 for VM are listed in Figure 39.

*Figure 39. TCP/IP V2 R4 for VM CATALOG Files*

Catalog File	TCP/IP for VM Feature
5735FALK CATALOG	TCP/IP V2 R4 for VM Base
5735FALL CATALOG	TCP/IP V2 R4 for VM NFS Feature
5735FALN CATALOG	TCP/IP V2 R4 for VM Kerberos US Feature
5735FALP CATALOG	TCP/IP V2 R4 for VM Kerberos WT Feature

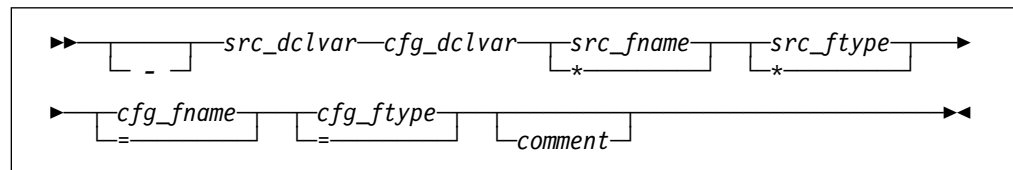
Distinct sections with the catalog are used for “product” files, such as the TCPIP MODULE, and “sample” files which need to be customized, such as the TCPIP SDATA file. Each section is defined via a pair of “begin” and “end” tags. For example, the section for the TCP/IP V2 R4 for VM Base sample configuration files is defined using the :TCPCONFIG. and :ETCPCONFIG. tags.

The entries within each section provide information sufficient for TCP2PROD to process a group of files associated with a given TCP/IP for VM feature.

Unless marked otherwise, all records within a given section will be processed, and the corresponding files placed into production via the VMSES/E VMFCOPY command.

### A.2.2 CATALOG Record Syntax

The syntax for TCP/IP for VM CATALOG file records is indicated below.



## TCP/IP for VM CATALOG Files

### A.2.3 Operands

- is the record bypass character, a hyphen (-). Use this bypass character at the beginning of record to indicate that this record (or file) should not be processed by TCP2PROD.  
**Note:** It's recommended that all file entries remain present within the CATALOG files. If a TCP/IP for VM file is not used in your environment, its corresponding entry should be bypassed as described above. Doing so will also help readily distinguish records corresponding to TCP/IP for VM product files from comment lines.
- src\_dclvar* is a PPF :DCL. variable name for the minidisk or SFS directory where the source file resides.
- cfg\_dclvar* is a PPF :DCL. variable name for the minidisk or SFS directory where the configured file is to reside.
- src\_fname* is the source file name. An asterisk (\*) can be used to indicate that all files of the type specified by *src\_ftype* are to be processed. When the \* wildcard is used, the configured file name remains unchanged from the source file name.
- src\_ftype* is the source file type. An asterisk (\*) can be used to indicate that all files of the name specified by *src\_fname* are to be processed. When the \* wildcard is used, the configured file type remains unchanged from the source file type.
- cfg\_fname* is the configured file name. If the source file name is specified using an asterisk (\*), *cfg\_fname* must be specified to maintain the correct record format. In such cases, an equals sign (=) should be used. Note that a value other than "=" will be processed as if the equals sign was specified.
- cfg\_ftype* is the configured file type. If the source file type is specified using an asterisk (\*), *cfg\_ftype* must be specified to maintain the correct record format. In such cases, an equals sign (=) should be used. Note that a value other than "=" will be processed as if the equals sign was specified.
- comment* is commentary text, which is ignored by TCP2PROD during processing.

#### Notes:

1. All parameters must be separated by at least one space.
2. Comment lines may be included in CATALOG files. Comment lines must begin with an asterisk (\*); these lines are ignored during TCP2PROD processing.

### A.2.4 Usage Notes

- If you modify a TCP/IP for VM CATALOG file, changes should be made via the VMSES/E local modification process so that any service-related changes can be reported during the VMSES/E service process. See Appendix B, “Modifying TCP/IP for VM CATALOG Files” on page 178 for more information about how to change TCP/IP for VM CATALOG files in this manner.
- The source and target minidisk/directory variable names used within this file correspond to those used in the applicable TCP/IP for VM feature PPF file (or an override variation of that file). If any changes are made to the Variable Declarations (:DCL.) section of a TCP/IP for VM PPF file via a PPF override, you may need to incorporate similar changes within the CATALOG files (via a separate local modification), so that PPF :DCL. variable names are correctly resolved.

---

## Appendix B. Modifying TCP/IP for VM CATALOG Files

This appendix describes how to create a VMSES/E local modification for a TCP/IP for VM Base or feature CATALOG file, which is a replacement-maintained part. Files for which this process is applicable are listed in Figure 39 on page 175.

For more information about installing and maintaining local modifications, see Chapters 5, 6, and 7 of the *VM/ESA Service Guide (SC24-5749)*.

### Notes:

1. Throughout this procedure, references are made to several TCP/IP for VM installation minidisks. Default addresses, and equivalent SFS directories (when applicable), for these minidisks are listed in the tables in 5.3, "DASD Storage and User ID Requirements" on page 20.
2. VMSES/E local modifications require a modification number, or *modid*, to be associated with the parts affected by a change; *modid* is a locally-determined value. It should begin with **L**, and is followed by up to 4 alphanumeric characters that identify a specific local modification. For example, L0001.

**1** Log on the installation user ID, **P735FALK**.

**2** Establish read access to the VMSES/E code and the Software Inventory minidisks.

**link maint 5e5 5e5 rr**

The 5E5 minidisk contains the VMSES/E code.

**link maint 51d 51d rr**  
**access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

**3** Create a usable PPF file.

**Note:** This VMFPPF step is necessary **only** if you have made changes to the PPF file since it was last compiled.

**vmfppf ppfname compname**



where

- *ppfname* is the name of the TCP/IP for VM feature PPF file with which the CATALOG file to be modified is associated. For example, **5735falk**. If you have created an override for the PPF file in question, use your override file name.
- *compname* is the feature component name with which the CATALOG file is associated. For example, for the TCP/IP for VM Base (5735FALK), you would use one of the following: **tcpip**, **tcpipsfs**, **tcpipnoc**, or **tcpipsfsnoc**

#### 4 Establish the correct minidisk access order.

**vmfsetup** *ppfname compname*

where *ppfname* and *compname* are the same as described in step 3 on page 178.

#### 5 Copy the CATALOG file to be modified to the local modification (LOCALMOD) minidisk.

**copyfile** *catalog-fn catalog sfm = catmodid fm-local*

where:

- *catalog-fn* is the file name of the CATALOG file being modified.
- *sfm* is the file mode of the minidisk where the CATALOG file resides.
- **catmodid** is the CATALOG file type abbreviation (CAT) concatenated with your local modification number, *modid* (*modid* is described in note 2 on page 178).
- *fm-local* is the file mode of the TCP/IP for VM local modification minidisk.

#### 6 Xedit the modification-level copy of the CATALOG file, then make and file your changes.

For example, if you choose to not use the NCS interface with TCP/IP for VM, you would not define the NCS, NCSLLBD, and NCSGLBD servers. To avoid TCP2PROD processing of the files associated with these servers, you would need to bypass the 5735FALK CATALOG file entries that follow, as indicated:

```

:TCPRUN.
...
- &BLD7Z &DISK13 * * = = NCS (195)
...
- &BLD1Z &DISK12 NCSPRO EXEC PROFILE EXEC NCS
- &BLD3Z &DISK12 NCSSOCK1 H SOCKET H NCS
- &BLD3Z &DISK12 NCSRPC H RPC H NCS
- &BLD3Z &DISK12 NCSUTIL C UTIL C NCS
- &BLD1Z &DISK14 NCSLLPRO EXEC PROFILE EXEC NCSLLBD
- &BLD1Z &DISK15 NCSGLPRO EXEC PROFILE EXEC NCSGLBD
...
:ETCPRUN.
...
:TCPCONFIG.
...
- &BLD0Z &BLD4Z NCSLLXIT EXEC = = NCSLLBD
- &BLD0Z &BLD4Z NCSGLXIT EXEC = = NCSGLBD
...
:ETCPCONFIG.

```

**xedit** *catalog-fn catmodid fm-local*

**make** *your required changes*

**==>** **file** *(to save the modified file)*

**7** Update the local version vector table (VVTLCCL) to create a record of your modification. The VVTLCCL will be created if none currently exists.

**vmfsim logmod** *appid vvtlcl fm-local tdata :part catalog-fn cat :mod lcmodid*

where *appid* is the appropriate TCP/IP for VM component APPID string, as follows:

**5735FALK** for the TCP/IP for VM Base  
**5735FALL** for the NFS Feature  
**5735FALN** for the Kerberos US Feature  
**5735FALP** for the Kerberos WT Feature

**8** Xedit the select data (\$SELECT) file on the alternate apply (APPLY) minidisk.

**Note:** This file may not currently exist on this minidisk.

```
| xedit appid $select fm-applyalt (noprof
| ==> top
| ==> input :APPLYID. mm/dd/yy hh:mm:ss
| ==> input catalog-fn cat
| ==> file
```

where

- *appid* is the appropriate TCP/IP for VM component APPID string, as described in step 7.
- *fm-applyalt* is the file mode of the alternate apply (APPLY) minidisk.
- *mm/dd/yy hh:mm:ss* are the (approximate) date and time at which you have made your modification.

**Note:** The date and time you provide **must be unique**.

## 9 Build the serviced CATALOG file.

```
| vmfbld ppf ppfname compname buildlist-fn catalog-fn.catalog (serviced
```

where:

- *buildlist-fn* is the appropriate TCP/IP for VM feature build list file name, as follows:
  - TCPBL491** for the TCP/IP for VM Base
  - NFSBL491** for the NFS Feature
  - USKBL491** for the Kerberos US Feature
  - WTKBL491** for the Kerberos WT Feature
- *catalog-fn* is the file name of the CATALOG file that you have modified.

---

## / Appendix C. Copying TCP/IP for VM Client Code to the / Y-Disk

/ To simplify access to TCP/IP client functions for your user community, you may find  
/ it desirable to copy all, or a subset of, the TCP/IP for VM client code to the  
/ VM/ESA product code disk. Doing so will avoid the need for users to additionally  
/ link and access the TCPMAINT 592 minidisk.

/ As well, applications that use certain programming interfaces may require  
/ TCP/IP-specific information to be available for proper operation. For example,  
/ information defined in the TCPIP DATA file is referenced by various functions  
/ provided as part of the CMS REXX Socket library. See the *VM/ESA REXX/VM*  
/ *Reference* (SC24-5770) for more information.

/ To copy TCP/IP V2 R4 for VM client files to the Product Code minidisk (typically  
/ the MAINT 19E minidisk — the Y-disk), use the following procedure **after** you have  
/ installed TCP/IP for VM and *all* TCP/IP features used in your environment.

### — Warning - File Overlap Considerations —

**Before** you copy *any* TCP/IP V2 R4 for VM client files to the Y-disk (or a similar  
“*common use*” minidisk), you should first determine whether any conflicts exist  
between the TCP/IP client files you choose to copy, and those present on the  
target (Y-disk) minidisk. If any file conflicts are found, these should be  
addressed and resolved with respect to your installation environment before you  
continue with the procedure that follows.

### Notes:

1. You will need to repeat this procedure each time you apply service to TCP/IP V2 R4 for VM or its features.
2. Use discretion when wildcards (\*) are used for both the *fn* (file name) and *ft* (file type) parameters of the VMFCOPY commands shown in this section, since files that exist on the Y-disk can be replaced with similarly-named TCP/IP counterpart files. The overlay of certain files may be warranted in some cases, and may be undesirable for others.

An example of this latter case is cited here. Both TCP/IP V2 R4 for VM and the IBM Language Environment for MVS and VM Release 5 have several **H** files that are identically named, but differ in content. These files are:

FCNTL	H	IF	H	IN	H	INET	H
IOCTL	H	NETDB	H	RESOURCE	H	RTROUTEH	H
SOCKET	H	STRINGS	H	TTYDEV	H	TYPES	H
UIO	H						

/ An overlay of the IBM Language Environment for MVS and VM Release 5 H  
/ files (already present on the Y-disk) by their TCP/IP counterparts may create  
/ problems when applications that expect (and rely upon) the content of the IBM  
/ Language Environment for MVS and VM Release 5 files are developed or  
/ rebuilt.

/ **1** Log on the **MAINT** (or equivalent) user ID.

/ **2** Process TCP/IP for VM files used by or available to TCP/IP clients.

/ **link tcpmaint 592 592 rr**  
/ **access 592 e**  
/ **access 19e f**

**Note:** If the Y-disk is not defined as the 19E  
minidisk in your environment, substitute the  
appropriate address for this minidisk.

/ **vmfcopy *fn ft e = = f2* (olddate replace sprodid 5735falk%*compname* prodid 5735falk%*compname***

/ where *compname* is the appropriate component  
/ name you used throughout the TCP/IP for  
/ VM installation or service process. For example,  
/ use **tcpip** or **tcpiptsfs** if you installed to minidisks  
/ or the Shared File System respectively, and you  
/ are using services that require Language  
/ Environment runtime library support. Otherwise,  
/ use **tcpiptnoc** (minidisks) or **tcpiptsfnoc** (SFS).

/ The VMFCOPY command will update the  
/ VMSES/E PARTCAT file on the Y-disk.

/ Wildcards (\*) can be substituted for *fn* (file name)  
/ and *ft* (file type), but should be used with  
/ discretion.

/ **3** Process files associated with each TCP/IP for VM feature that you have  
/ installed or serviced.

/ 

- Process NFS Feature files used by TCP/IP for VM clients.

/ **vmfcopy *fn ft e = = f2* (olddate replace sprodid 5735fall%*compname* prodid 5735fall%*compname***

/ where *compname* is the appropriate component  
 / name you used used throughout the NFS Feature  
 / installation or service process. For example, use  
 / **vmnfs** or **vmnfssfs**, if you installed to minidisks or  
 / the Shared File System respectively.  
 /  
 / The VMFCOPY command will update the  
 / VMSES/E PARTCAT file on the Y-disk.  
 /  
 / Wildcards (\*) can be substituted for *fn* (file name)  
 / and *ft* (file type), but should be used with  
 / discretion.

- Process Kerberos US Feature files used by TCP/IP for VM clients.

/ **vmfcopy *fn ft e = = f2 (olddate replace sprodid 5735faln%compname prodid 5735faln%compname***

/ where *compname* is the appropriate component  
 / name you used used throughout the Kerberos US  
 / Feature installation or service process. For  
 / example, use **vmkerb** or **vmkerbsfs**, if you  
 / installed to minidisks or the Shared File System  
 / respectively.  
 /  
 / The VMFCOPY command will update the  
 / VMSES/E PARTCAT file on the Y-disk.  
 /  
 / Wildcards (\*) can be substituted for *fn* (file name)  
 / and *ft* (file type), but should be used with  
 / discretion.

- Process Kerberos WT Feature files used by TCP/IP for VM clients.

/ **vmfcopy *fn ft e = = f2 (olddate replace sprodid 5735falp%compname prodid 5735falp%compname***

/ where *compname* is the appropriate component  
 / name you used used throughout the Kerberos WT  
 / Feature installation or service process. For  
 / example, use **vmkerb** or **vmkerbsfs**, if you  
 / installed to minidisks or the Shared File System  
 / respectively.  
 /  
 / The VMFCOPY command will update the  
 / VMSES/E PARTCAT file on the Y-disk.  
 /  
 / Wildcards (\*) can be substituted for *fn* (file name)  
 / and *ft* (file type), but should be used with  
 / discretion.

/

/

/

/

**4** (Optional) Erase any TCP/IP for VM files that you do not want on the Y-disk — for example, any MAP files that correspond to TCP/IP for VM modules re-built during service. Refer to the VMSES/E PARTCAT file on Y-disk to determine which files are associated with TCP/IP for VM.

/

/

/

/

**Note:** Additional information about the various TCP/IP for VM client files, and their association with specific TCP/IP functions, is available on-line via the TCP/IP for VM home page on the World Wide Web. The URL for this home page is:

/

<http://www.ibm.com/s390/vm/related/tcpip/>

/ **vmferase file filename filetype f**

/

/

/

See the *VMSES/E Introduction and Reference* for more information about the VMFERASE command and options that may help you remove specific files.

/

/

/

/

**5** Re-save the CMS saved system, to return the Y-disk to shared status. See the “Placing (Serviced) Components into Production” section of the *VM/ESA Service Guide* for detailed information about how to save the CMS saved system.

## Appendix D. TCP/IP for VM Build Lists

This appendix provides a complete list of the VMSES/E build lists used to maintain the TCP/IP Version 2 Release 4 for VM product. This information has been provided to help you determine which build list to use with VMSES/E commands when you need to build or service specific TCP/IP for VM objects, and assist you with making local modifications. For more information about build list content and formats, see the *VMSES/E Introduction and Reference* (SC24-5747 or SC24-5444).

The build lists identified in the tables that follow can be found on the P735FALK 2B2 (BASE1) minidisk. However, before using the information in a build list, the P735FALK 2D2 (DELTA) minidisk should be checked to determine whether that build list has been updated by service, so that the most current file is referenced.

Also, note that the minidisks shown under the "Build String" headings are default values for P735FALK minidisks. If a PPF override has been used in your environment to change Build String minidisks or SFS directories, use your values when determining which files are affected by a build list.

### D.1 TCP/IP V2 R4 for VM Build Lists

The Figure 40 lists the VMSES/E build lists used with the TCP/IP V2 R4 for VM Base product, and provides general information about the objects (files) managed by each build list.

Figure 40 (Page 1 of 2). VMSES/E Build Lists - TCP/IP V2 R4 for VM

Build List Name	VMSES/E Part Handler	Build String (Minidisk)	Build List Description / Affected Objects
TCPBL491	VMFBDCOM	BUILD1 (491)	Full-replacement objects built to the 491 minidisk
TCPBL492	VMFBDCOM	BUILD3 (492)	Full-replacement objects built to the 492 minidisk
TCPBL493	VMFBDCOM	BUILD5 (493)	Full-replacement objects built to the 493 minidisk
TCPBLTRP	VMFBMLB	BUILD3 (492)	TFTPRP MACLIB build list
TCPBLALL	VMFBMLB	BUILD3 (492)	ALLMACRO MACLIB build list
TCPBLCOM	VMFBTLB	BUILD3 (492)	COMMTXT TXTLIB build list
TCPBLGDD	VMFBTLB	BUILD3 (492)	GDDMXD TXTLIB build list
TCPBLXAW	VMFBTLB	BUILD3 (492)	XAWLIB TXTLIB build list
<b>Notes:</b>			
1. Language Environment runtime library support must be available when building objects identified in this build list.			



Figure 40 (Page 2 of 2). VMSES/E Build Lists - TCP/IP V2 R4 for VM

Build List Name	VMSES/E Part Handler	Build String (Minidisk)	Build List Description / Affected Objects
TCPBLDPI	VMFBDTLB	BUILD3 (492)	DPILIB TXTLIB build list
TCPBLRPC	VMFBDTLB	BUILD3 (492)	RPCLIB TXTLIB build list
TCPBLOLD	VMFBDTLB	BUILD3 (492)	OLDXLIB TXTLIB build list
TCPBLXTL	VMFBDTLB	BUILD3 (492)	XTLIB TXTLIB build list
TCPBLX11	VMFBDTLB	BUILD3 (492)	X11LIB TXTLIB build list
TCPBLSNA	VMFBDLLB	BUILD1 (491)	SNALINK LOADLIB build list
TCPBLXNX	VMFBDLLB	BUILD1 (491)	XNX25 LOADLIB build list
TCPBLSNM	VMFBDLLB	BUILD1 (491)	SNMPLIB LOADLIB build list
TCPBLM91	VMFBDMOD	BUILD1 (491)	MODULE objects built to the 491 minidisk
TCPBLM92	VMFBDMOD	BUILD3 (492)	MODULE objects built to the 492 minidisk
TCPBLC91 (1*)	VMFBDMOD	BUILD1 (491)	C-based MODULE objects built to the 491 minidisk
TCPBLC92 (1*)	VMFBDMOD	BUILD3 (492)	C-based MODULE objects built to the 492 minidisk
TCPBLCN5 (1*)	VMFBDMOD	BUILD7 (395)	C-based MODULE objects built to the 395 minidisk
<b>Notes:</b>			
1. Language Environment runtime library support must be available when building objects identified in this build list.			

## D.2 NFS Feature Build Lists

The Figure 41 lists the VMSES/E build lists used with the NFS Feature, and provides general information about the objects (files) managed by each build list.

Figure 41. VMSES/E Build Lists - NFS Feature

Build List Name	VMSES/E Part Handler	Build String (Minidisk)	Build List Description / Affected Objects
NFSBL491	VMFBDCOM	BUILD1 (491)	Full-replacement objects built to the 491 minidisk
NFSBL492	VMFBDCOM	BUILD3 (492)	Full-replacement objects built to the 492 minidisk
NFSBLC91 (1*)	VMFBDMOD	BUILD1 (491)	C-based MODULE objects built to the 491 minidisk
<b>Notes:</b>			
1. Language Environment runtime library support must be available when building objects identified in this build list.			

## D.3 Kerberos US Feature Build Lists

The Figure 42 lists the VMSES/E build lists used with the Kerberos US Feature, and provides general information about the objects (files) managed by each build list.

Figure 42. VMSES/E Build Lists - Kerberos US Feature

Build List Name	VMSES/E Part Handler	Build String (Minidisk)	Build List Description / Affected Objects
USKBL491	VMFBDCOM	BUILD1 (491)	Full-replacement objects built to the 491 minidisk
USKBL492	VMFBDCOM	BUILD3 (492)	Full-replacement objects built to the 492 minidisk
USKBLDES	VMFBDTLB	BUILD3 (492)	DES TXTLIB build list
USKBLKRB	VMFBDTLB	BUILD3 (492)	KRB TXTLIB build list
USKBLKDB	VMFBDTLB	BUILD3 (492)	KDB TXTLIB build list
USKBLBPL	VMFBDTLB	BUILD3 (492)	BPLDBM TXTLIB build list
USKBLC91 (1*)	VMFBDMOD	BUILD1 (491)	C-based MODULE objects built to the 491 minidisk
USKBLC92 (1*)	VMFBDMOD	BUILD3 (492)	C-based MODULE objects built to the 492 minidisk

**Notes:**

1. Language Environment runtime library support must be available when building objects identified in this build list.

## D.4 Kerberos WT Feature Build Lists

Figure 43 lists the VMSES/E build lists used with the Kerberos WT Feature, and provides general information about the objects (files) managed by each build list.

Figure 43 (Page 1 of 2). VMSES/E Build Lists - Kerberos WT Feature

Build List Name	VMSES/E Part Handler	Build String (Minidisk)	Build List Description / Affected Objects
WTKBL491	VMFBDCOM	BUILD1 (491)	Full-replacement objects built to the 491 minidisk
WTKBL492	VMFBDCOM	BUILD3 (492)	Full-replacement objects built to the 492 minidisk
WTKBLKRB	VMFBDTLB	BUILD3 (492)	KRB TXTLIB build list
WTKBLKDB	VMFBDTLB	BUILD3 (492)	KDB TXTLIB build list

**Notes:**

1. Language Environment runtime library support must be available when building objects identified in this build list.

Figure 43 (Page 2 of 2). VMSES/E Build Lists - Kerberos WT Feature

<b>Build List Name</b>	<b>VMSES/E Part Handler</b>	<b>Build String (Minidisk)</b>	<b>Build List Description / Affected Objects</b>
WTKBLBPL	VMFBDTLB	BUILD3 (492)	BPLDBM TXTLIB build list
WTKBLC91 (1*)	VMFBDMOD	BUILD1 (491)	C-based MODULE objects built to the 491 minidisk
WTKBLC92 (1*)	VMFBDMOD	BUILD3 (492)	C-based MODULE objects built to the 492 minidisk
<p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Language Environment runtime library support must be available when building objects identified in this build list.</li> </ol>			

---

## Appendix E. IBM Network Station Support

**Note!**

This appendix consolidates information previously found only in the TCP/IP V2 R4 for VM Corrective Service Memo.

Be certain to review the information in this appendix before you install the service identified in E.1, "IBM Network Station Support Overview."

---

### E.1 IBM Network Station Support Overview

This appendix describes the TCP/IP V2 R4 for VM support for the IBM Network Station, which is provided through the following corrective service:

<b>APAR (PTF number)</b>	<b>Description</b>
<b>PQ01770</b> (UQ03096)	Network Station Support - BOOTPD / TFTPDP Function
<b>PQ02301</b> (UQ03142)	Network Station Support - Client Code Function
<b>PQ04280</b> (UQ05061)	Network Station Support - TCPRUN BFS Directory / README Updates

Before this service can be applied to your system, additional resources must be allocated for TCP/IP for VM. Significant among the changes required are the need to:

**Define additional user IDs and minidisks**  
**Increase DELTA (2D2) minidisk storage**

**Notes:**

1. The TCP/IP V2 R4 for VM BOOTPD and TFTPDP server functions, available through APAR PQ01770, are provided specifically to support the IBM Network Station.
2. The IBM Network Station support provided by the corrective service identified above requires VM/ESA Version 2 Release 1.0, or later, and the Byte File System (BFS) support provided by these releases of VM/ESA.

A complete list of files added or changed with the above service can be found in E.2.8, "New and Changed Files" on page 195.

---

## E.2 IBM Network Station Support Requirements

This section describes additional TCP/IP for VM requirements that must be in effect before you can apply the corrective service identified in E.1, “IBM Network Station Support Overview” on page 190.

Information about how to configure the BOOTPD and TFTP servers, and how to process the IBM Network Station “Client Code,” is provided in the following appendices:

- Appendix H, “Configuring the BOOTPD Virtual Machine” on page 204
- Appendix J, “Configuring the TFTP Virtual Machine” on page 314
- Appendix K, “Processing the IBM Network Station Client Code” on page 333

### E.2.1 Operating System Requirements

Before you install TCP/IP V2 R4 for VM support for the IBM Network Station, the following CMS service must be installed:

- For VM/ESA Version 2 Release 1.0:  
VM61080 (PTF UM28330)
- For VM/ESA Version 2 Release 2.0:  
VM61080 (PTF UM28331)

### E.2.2 DASD Storage Requirements

TCP/IP for VM support for the IBM Network Station requires additional minidisks to be defined, and the size of some existing minidisks to be increased. See 5.3.3, “DASD Storage Requirements for IBM Network Station Support” on page 28 for detailed information about these requirements.

**Note**

All new minidisks that you add for IBM Network Station support must be formatted before you install the corrective service provided by PTF UQ03096.

### E.2.3 Byte File System (BFS) Storage Requirements

The Byte File System (BFS) is used to maintain the various files that the TFTP daemon loads down to the IBM Network Stations. These files (the “Client Code”), are provided via APAR PQ02301 as a system “tar” file, NSTATION TARBIN. You will need to ensure sufficient BFS file space is available to accept these files when the NSTATION TARBIN file is “exploded,” or processed. The space required for the resulting “Client Code” files is approximately **8000** 4K blocks.

## E.2.4 User ID Requirements

TCP/IP for VM support for the IBM Network Station requires two additional user IDs to be defined — **BOOTPD** and **TFTPD**. Sample directory entries for these user IDs follow:

```
USER BOOTPD BOOTPD 32M 32M G
INCLUDE CMSUSER
MACHINE XC
OPTION QUICKDSP SVMSTAT
LINK TCPMAINT 591 591 RR
LINK TCPMAINT 592 592 RR
LINK TCPMAINT 198 198 RR
```

```
USER TFTPD TFTPD 64M 64M G
INCLUDE CMSUSER
POSIXINFO UID 0 GID 0
MACHINE XC
OPTION QUICKDSP SVMSTAT APPLMON
LINK TCPMAINT 591 591 RR
LINK TCPMAINT 592 592 RR
LINK TCPMAINT 198 198 RR
```

## E.2.5 Changed User ID Requirements

The P735FALK user ID directory entry and PROFILE EXEC must be modified to account for several requirements associated with IBM Network Station support. These changes are described in the following sections.

### E.2.5.1 P735FALK Directory Entry Requirements

The P735FALK directory entry changes required include:

- additional LINK statements to new product minidisks, identified in Figure 21 on page 28
- a LINK statement for the MAINT 193 minidisk
- increased virtual storage, to allow for processing of the IBM Network Station Client Code
- adding a POSIXINFO statement for POSIX “superuser” capability

See the *OpenEdition for VM/ESA User's Guide* and *VM/ESA Planning and Administration* for more information about defining user IDs as POSIX “superusers.”

An abbreviated sample P735FALK directory entry which highlights the changes previously described follows:

```
USER P735FALK P735FALK 16M 64M BG
INCLUDE CMSUSER
POSIXINFO UID 0 GID 0
...
LINK BOOTPD 191 273 MR
LINK TFTP 191 275 MR
LINK MAINT 193 193 RR
...
```

### E.2.5.2 Other P735FALK user ID Requirements

To allow for processing of the IBM Network Station Client Code files, the P735FALK user ID must be given administrator authority for the BFS file pool in which these files are installed. See *VM/ESA File Pool Planning, Administration and Operation* for details about SFS/BFS file pool configuration and administrator authority.

To ensure that the TFTP server module will correctly build, verify that the **DMSPSLU TXTLIB**, provided by APAR VM61080, is available. The following command can be used to accomplish this:

```
global txtlib dmsslu
```

Changes to the P735FALK PROFILE EXEC are necessary as well, so that the CMS **LOADBFS EXEC**, used to process the NSTATION TARBIN file, will be present in the CMS search order. To accomplish this, update the PROFILE EXEC to include the following statement:

```
'ACCESS 193' fm
```

where *fm* is an available file mode, for example, 0.

## E.2.6 Server Run-time and Configuration Files

Several new server-related files are provided with the IBM Network Station support. These files are indicated in Figure 44 on page 194 and Figure 45 on page 194. As with other product and configuration files, these files are processed by the TCP2PROD exec in conjunction with the 5735FALK CATALOG file. An updated 5735FALK CATALOG is provided with APAR PQ01770 which will process these new files.

Figure 44 on page 194 lists the name and location of the server runtime files provided by APAR PQ01770, and the names and location of these files after they've been placed into production. The servers which use a given file have been identified as well.

Figure 44. Files to Copy into Production - IBM Network Station Support

P735FALK Test Disk	P735FALK Link Address	Original File Name/Type	Production File Name/Type	Server Minidisk
491	273	BOOTPPRO EXEC	PROFILE EXEC	BOOTPD 191
491	275	TFTPDPRO EXEC	PROFILE EXEC	TFTPD 191

Figure 45 lists the name and location of the sample files provided by APAR PQ01770, and the names and location of these files after they've customized. The servers which use a given file have been identified as well.

Figure 45. Files to Customize - IBM Network Station Support

Sample Disk	Configured Disk	Sample File Name/Type	Configured File Name/Type	Server
591	198	BOOTPXIT EXEC	no change	BOOTPD
591	198	BOOTPTAB SAMPLE	ETC BOOTPTAB	BOOTPD
591	198	TFTPDXIT EXEC	no change	TFTPD

## E.2.7 Service Considerations

### E.2.7.1 Maintaining Multiple NSTATION TARBIN (TARnnnnn) Files

The increased DASD storage requirements for the P735FALK 2D2 minidisk, identified in Figure 22 on page 29 of 5.3.3.2, "DASD Storage Requirements for Existing Minidisks" on page 29, are sufficient to maintain two distinct PTF-level copies of the NSTATION TARBIN file. (The separate levels of this file are maintained as NSTATION TARnnnnn, where nnnnn is a PTF number. For example, NSTATION TAR05250.) If you want to maintain more than two such copies of this file, you will need to increase the P735FALK 2D2 storage accordingly.

### E.2.7.2 Pre-Service Processing

Before you receive corrective service that affects the NSTATION TARBIN file, you need to ensure sufficient 2D2 minidisk space is available to receive an updated copy of this file. This is especially true as the NSTATION TARBIN file is serviced over time.

Assuming that the P735FALK 2D2 minidisk has been sufficiently increased to maintain two copies of this file at a given time, you will need to erase the **oldest** PTF-level copy of this file from this minidisk. Use the following command to list the various PTF-level copies of the NSTATION TARBIN file:

```
filelist nstation tar* fm-delta
```



where *fm-delta* is the file mode of the P735FALK 2D2 minidisk. After you have erased the appropriate NSTATION TARnnnnn file, you then can continue the service process with the step, 7.2.3.2, "Receive the Service" on page 128.

## E.2.8 New and Changed Files

Files which have been added to TCP/IP for VM via APAR PQ01770 are listed below. These new files provide the BOOTPD and TFTPDP function necessary to support the IBM Network Station.

### E.2.8.1 New Files - PQ01770

- BOOTPD Files

```
BOOTPD $EXEC/EXEC BOOTPD HELP
BOOTPPRO $EXEC/EXEC BOOTPD README
BOOTPXIT $EXEC/EXEC
BOOTPTAB SAMPLE
```

- TFTPDP Files

```
TFTPDPRO $EXEC/EXEC TFTPDP HELP
TFTPDXIT $EXEC/EXEC TFTPDP README
TFTPDP MODULE
```

The TFTPDP MODULE (built locally) is comprised of the following:

```
TFTCMD TEXT TFTCON TEXT TFTDFL TEXT TFTHLP TEXT
TFTINI TEXT TFTIPM TEXT TFTLFL TEXT TFTMAIN TEXT
TFTPFL TEXT TFTQRY TEXT TFTRFL TEXT TFTTIM TEXT
TFTTOK TEXT TFTTRC TEXT TFTTRM TEXT TFTUID TEXT
TFTWFL TEXT TFTXLT TEXT
```

- TFTPDP MACLIB (built locally)

```
TFTANCBK COPY TFTBIND COPY TFTCDBK COPY TFTCLIBK COPY
TFTCONST COPY TFTDBKBK COPY TFTDGRBK COPY TFTFILBK COPY
TFTMONBK COPY TFTPFLBK COPY TFTSRPBK COPY TFTTIMBK COPY
TFTUIDBK COPY TFTXMIBK COPY
```

- IBM Network Station - Related Files

```
NSTATION TARBIN ("dummy" file only)
NSTATION README
NSTATION $LOADBFS/LOADBFS
```

- Message Repositories

```
TCPUME REPOS/TEXT
TCPUMB REPOS/TEXT
```

- VMSES/E-related Files

```
TCPBLTRP $EXEC/EXEC
TCPBL493 $EXEC/EXEC
```

### E.2.8.2 New Files - PQ02301

Files which have been added to TCP/IP for VM via APAR PQ02301 are listed below. These new files provide the IBM Network Station Client Code files, which run within the IBM Network Station, and information about installing these files in the Byte File System (BFS).

- IBM Network Station - Related Files

NSTATION TARBIN

### E.2.8.3 Changed Files - PQ01770

Files which have been changed via APAR PQ01770 are listed below.

- Server-related Files

TCPRUN \$EXEC/EXEC

- VMSES/E-related Files

5735FALK \$PPF TCPBLM91 \$EXEC/EXEC

5735FALK PRODPART TCPBL491 \$EXEC/EXEC

5735FALK CATALOG

### E.2.8.4 Changed Files - PQ04280

Files which have been changed via APAR PQ04280 are listed below.

- Server-related Files

TCPRUN \$EXEC/EXEC

- IBM Network Station - Related Files

NSTATION README

BOOTPD README

TFPTD README

---

## / Appendix F. DHCPD Server Support

/

/

/

/

**Note!**

Be certain to review the information in this appendix before you install the service identified in F.1, "DHCPD Server Support Overview."

---

### / F.1 DHCPD Server Support Overview

/ This appendix describes the TCP/IP V2 R4 for VM support for the DHCPD server, which is provided through the following corrective service:

<b>APAR (Initial PTF)</b>	<b>Description</b>
PQ05305 (UQ15557)	DHCPD Server Support
PQ13330 (UQ15381)	TCPIP Server Update for DHCPD Support

/ Before this service can be applied to your system, additional resources must be allocated for TCP/IP for VM. Significant among the changes required are the need to:

- Define an additional user ID and minidisk**
- Increase minidisk storage for various minidisks**

/

/

/

**Notes:**

1. The DHCPD server function, available through APAR PQ05305, is provided specifically to support the IBM Network Station.
2. This support requires VM/ESA Version 2 Release 1.0, or later.

/ A complete list of files added or changed with the above service can be found in F.2.6, "New and Changed Files" on page 200.

---

### / F.2 IBM Network Station Support Requirements

/ This section describes additional TCP/IP for VM requirements that must be in effect before you can apply the corrective service identified in F.1, "DHCPD Server Support Overview."

/ Information about how to configure the DHCPD server is provided in:

- Appendix I, "Configuring the DHCPD Virtual Machine" on page 226

/ **F.2.1 Operating System Requirements**

/ Before you install TCP/IP V2 R4 for VM support for the DHCPD server, the  
/ following CMS service must be installed:

- / • For VM/ESA Version 2 Release 1.0:  
/ VM61480 (Initial PTF: UM28656)
- / • For VM/ESA Version 2 Release 2.0:  
/ VM61480 (Initial PTF: UM28657)

/ **F.2.2 DASD Storage Requirements**

/ TCP/IP for VM support for the DHCPD server requires additional minidisks to be  
/ defined, and the size of some existing minidisks to be increased. See 5.3.4,  
/ "DASD Storage Requirements for DHCPD Server Support" on page 29 for detailed  
/ information about these requirements.

/ **Note**

/ All new minidisks that you add for IBM Network Station support must be  
/ formatted before you install the corrective service provided by PTF UQ15557.

/ **F.2.3 User ID Requirements**

/ TCP/IP for VM support for the DHCPD server requires one additional user ID to be  
/ defined — **DHCPD**. A sample directory entry for this user ID follows:

```
/ USER DHCPD DHCPD 64M 64M G  
/ INCLUDE CMSUSER  
/ MACHINE XC  
/ OPTION QUICKDSP SVMSTAT  
/ LINK TCPMAINT 591 591 RR  
/ LINK TCPMAINT 592 592 RR  
/ LINK TCPMAINT 198 198 RR
```

/ **F.2.4 Changed User ID Requirements**

/ The P735FALK user ID directory entry and PROFILE EXEC must be modified to  
/ account for several requirements associated with DHCPD server support. These  
/ changes are described in the following section.

/ **F.2.4.1 P735FALK Directory Entry Requirements**

/ The P735FALK directory entry changes required include:

- / • additional LINK statements to new product minidisks, identified in Figure 23 on  
/ page 29.

/ An abbreviated sample P735FALK directory entry, which highlights the changes  
/ previously described, follows:

```
/          USER P735FALK P735FALK 16M 64M BG
/          INCLUDE CMSUSER
/          ...
/          LINK DHCPD 191 278 MR
/          ...
```

## / **F.2.5 Server Run-time and Configuration Files**

/ Several new server-related files are provided with the DHCPD server support.  
/ These files are indicated in Figure 46 on page 200 and Figure 47 on page 200.  
/ As with other product and configuration files, these files are processed by the  
/ TCP2PROD exec in conjunction with the 5735FALK CATALOG file. An updated  
/ 5735FALK CATALOG is provided with APAR PQ05305, which will process these  
/ new files.

/ Figure 46 on page 200 lists the name and location of the server runtime files  
/ provided by APAR PQ05305, and the names and location of these files after  
/ they've been placed into production. The servers which use a given file have been  
/ identified as well.

<i>Figure 46. Files to Copy into Production - DHCPD Server Support</i>				
<b>P735FALK Test Disk</b>	<b>P735FALK Link Address</b>	<b>Original File Name/Type</b>	<b>Production File Name/Type</b>	<b>Server Minidisk</b>
491	278	DHCPDPRO EXEC	PROFILE EXEC	DHCPD 191

Figure 47 lists the name and location of the sample files provided by APAR PQ05305, and the names and location of these files after they've customized. The servers which use a given file have been identified as well.

<i>Figure 47. Files to Customize - DHCPD Server Support</i>				
<b>Sample Disk</b>	<b>Configured Disk</b>	<b>Sample File Name/Type</b>	<b>Configured File Name/Type</b>	<b>Server</b>
591	198	DHCPDXIT EXEC	no change	DHCPD

## F.2.6 New and Changed Files

Files which have been added to TCP/IP for VM via APAR PQ05305 are listed below. These new files provide the DHCPD server function.

### F.2.6.1 New Files - PQ05305

- DHCPD Files
  - DHCPD \$EXEC/EXEC DHCPD HELP
  - DHCPDPRO \$EXEC/EXEC DHCPD README
  - DHCPDXIT \$EXEC/EXEC

### F.2.6.2 Changed Files - PQ05305

Files which have been changed via APAR PQ05305 are listed below.

- Server-related Files
  - TCPRUN \$EXEC/EXEC
- Message Repositories
  - TCPUME REPOS/TEXT
  - TCPUMB REPOS/TEXT
- VMSES/E-related Files
  - 5735FALK \$PPF TCPBL491 \$EXEC/EXEC
  - 5735FALK PRODPART 5735FALK CATALOG

---

## Appendix G. Overriding the TFTP D Minidisk Link Address

This section provides information to help you create a product parameter file (PPF) override for the 5735FALK \$PPF file, which is updated with the corrective service for IBM Network Station support (APAR PQ01770—PTF UQ03096). This override is necessary to avoid a conflict between the P735FALK link address used for the (new) TFTP D 191 minidisk, and that used for the VMNFS 191 minidisk (if/when the TCP/IP for VM NFS Feature is installed).

When you apply the corrective service for PTF UQ03096, you will be notified during the step, 7.2.3.4, “Update the Build Status Table” on page 130, that the 5735FALK \$PPF file has been updated, and that all product parameter files built from this source file must be recompiled.

However, **BEFORE** you recompile the PPF files via the VMFPPF command, you need to create a PPF override for the **tcpip** component :DCL. section of the 5735FALK \$PPF file; this override will change the P735FALK link address for the TFTP D 191 minidisk. If you have an existing override for the TCP/IP V2 R4 for VM PPF file, you'll need to incorporate this change within your override.

**Note:** Do **not** directly modify (edit) the product supplied 5735FALK \$PPF or 5735FALK PPF files to change the :DCL. entry in question (or any other installation parameters). If the 5735FALK \$PPF file is further serviced, the existing \$PPF file will be replaced, and any changes to that file will be lost; by creating your own \$PPF override, your updates will be preserved.

The following process describes how to change the TFTP D minidisk link address from its current value of 274, to a non-conflicting value of 275.

- 1 Create a new \$PPF override file, or edit your existing override file.

**xedit** *overname* \$ppf *fm*2

*overname* is the PPF override file name (such as “tcpipnws”) that you want to use.

*fm* is an appropriate file mode. If you create this file yourself, specify a file mode of A.

If you modify an existing override file, specify a file mode of A or D, based on where the file currently resides (A being the file mode of a R/W 191 minidisk, or equivalent; D, that of the MAINT 51D minidisk).

- 2 Create (or modify as required) the Variable Declarations (:DCL.) section for the **tcPIP** component, so that it resembles the :DCL. section shown below. Since other TCP/IP for VM Base components override this section, these are included in the override list; corresponding overrides for these component areas are also included.

**Note!**

This override, after it is created, **must** be used for all service that is applied to TCP/IP V2 R4 for VM.

```

:OVERLST. TCPIP TCPIPSFS TCPIPNOC TCPIPSFSNOC
*****
* Override to change the TFTP 191 minidisk LINK address *
*****
:TCPIP. TCPIP 5735FALK
:DCL. UPDATE
  &DISK20 LINK TFTP 191 275 MR * TFTP A-disk
:EDCL.
:END.
*
:TCPIPSFS. TCPIP
:END.
*
:TCPIPNOC. TCPIP
:END.
*
:TCPIPSFSNOC. TCPIP
:END.

```

This override will update the :DCL. section for the **tcPIP**, **tcPipsfs**, **tcPIPnOC**, and **tcPipsfsnOC** components of the 5735FALK \$PPF file,

- 3 If your \$PPF override file was created at file mode A, copy it to file mode D, the Software Inventory minidisk (MAINT 51D).

```

copyfile overname $PPF fm = d (olddate)
erase overname $PPF a

```

- 4 Compile your changes to create the usable *overname* PPF file(s) for your environment.

```
vmfppf overname *
```

where *overname* is the file name of your \$PPF override file.





---

## Appendix H. Configuring the BOOTPD Virtual Machine

**Note!**

This appendix consolidates information previously found in the BOOTPD README file provided with APAR PQ01770, and the TCP/IP V2 R4 for VM Corrective Service Memo.

The BOOTPD virtual machine (daemon) responds to client requests for boot information using information contained in a BOOTP machine file.

To configure the BOOTPD virtual machine, you must perform the following steps:

**BOOTPD Configuration Steps**

1. Update the TCPIP server configuration file.
2. Update the BOOTPD profile exit.
3. Configure the ETC BOOTPTAB file.

This appendix describes the files used to configure BOOTPD server, as well as the BOOTPD command and subcommands which control its operation.

---

### H.1 Update the TCPIP Server Configuration File

You should include BOOTPD in the AUTOLOG statement of the TCPIP server configuration file (PROFILE TCPIP) so that the BOOTPD virtual machine will be automatically started when TCPIP is invoked. Verify that the following statements are added to the PROFILE TCPIP file:

```
AUTOLOG
  BOOTPD 0
```

The BOOTPD server listens on port 67. Verify that the following statement is added to your TCPIP server configuration file as well:

```
PORT
  67  UDP BOOTPD ; BOOTPD Server
```

---

## H.2 Update the BOOTPD Profile Exit

The BOOTPD PROFILE EXEC invokes the BOOTPXIT EXEC customization exit (if it exists) and stacks the TCPRUN EXEC for execution. TCPRUN then starts the BOOTPD server using the TCPRUN global variables OWNER, COMMAND, and PARMS. You can use the BOOTPXIT customization exit to alter these variables to change the console log owner, the server module name, or the parameters passed to the server module. See Chapter 5, “Server Profile Exits” of *TCP/IP V2 R4 for VM: Planning and Customization* for more information about using the customization exits.

**Note:** You must modify the BOOTPXIT EXEC if:

- You are using a server module that is not located on the TCP/IP server minidisk (TCPMAINT 591).
- You change parameters passed to the BOOTPD command.
- You change the user ID of the virtual machine that receives the BOOTPD console output.

**Note:** If no parameters are specified in the BOOTPXIT customization exit, BOOTPD is initialized with the following operands:

```
MACHINE ETC BOOTPTAB *
```

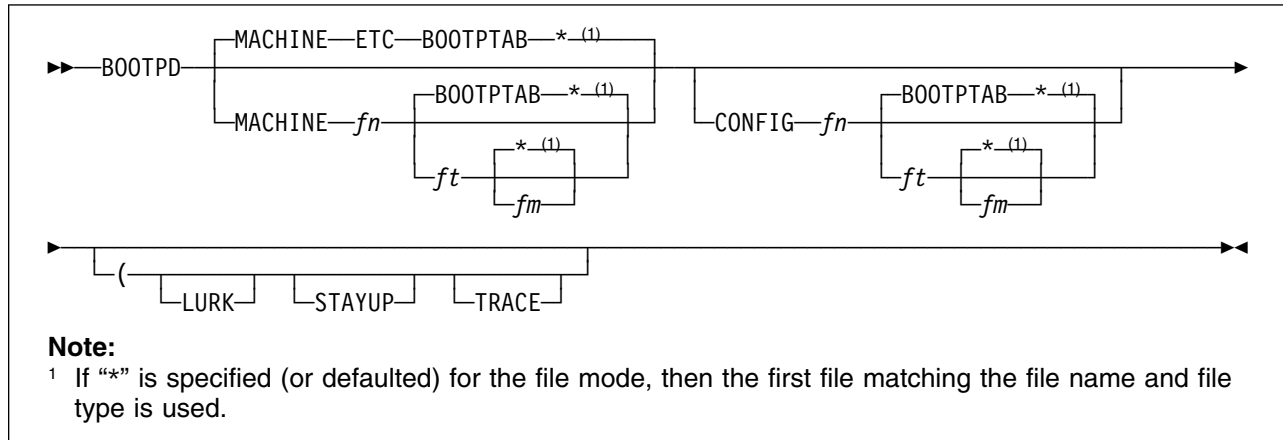
---

## H.3 Configure the ETC BOOTPTAB File

The BOOTPD server searches the ETC BOOTPTAB file for information when it attempts to satisfy BOOT requests generated by BootP clients on the network. The ETC BOOTPTAB file is located on the TCPMAINT 198 minidisk. (This file is initially created when files are copied into production during installation, and is a copy of the supplied BOOTPTAB SAMPLE file.) Before you use the BOOTPD server, you need to customize the ETC BOOTPTAB file with information about the IBM Network Stations that will be used in your environment. See the comments within the ETC BOOTPTAB file for detailed information about how to specify entries within this file. The customized ETC BOOTPTAB file should remain on the TCPMAINT 198 minidisk.

## BOOTPD Command

### H.4 BOOTPD Command



#### H.4.1 Purpose

The BootP daemon (the BOOTPD server) responds to client requests for boot information using information contained in a BOOTP machine file. This information includes the IP address of the client, the IP address of the TFTP daemon and information about the files to request from the TFTP daemon.

#### H.4.2 Operands

##### **MACHINE**

indicates that the file specification that follows specifies a file containing client information.

*fn* is the file name of the file to load.

*ft* is the file type of the file to load.

*fm* is the file mode of the file to load.

##### **CONFIG**

indicates that the file specification that follows specifies a file containing configuration information. This information lists adapters on the host which should be monitored and those that should not. Also, whether forwarding of BOOT requests should occur and when and where they should be sent.

##### **LURK**

indicates that the BootP daemon should never respond to a client request. It should only listen.

**STAYUP**

indicates that the BootP daemon should continue to operate across VM TCP/IP failures.

**TRACE**

indicates that the BootP daemon should display debug information as requests are processed.

**H.4.3 Usage Notes**

1. MACHINE and CONFIG are reserved keywords. They may not be used as file names or file types.
2. The defaults for the BOOTPD command when no operands or options are specified are:
  - “ETC BOOTPTAB \*\*” is the machine file,
  - LURK mode is disabled,
  - STAYUP mode is disabled,
  - TRACE mode is disabled,
  - No configuration file is used; thus, the BootP daemon will listen to BOOT requests received on any IP address.
3. The configuration file is composed of blank lines, comment lines, and configuration statements. Configuration statements have the same function and syntax as the BOOTPD **EXCLUDE**, **FORWARD**, and **INCLUDE** subcommands.

The format of the configuration file is:

- One line per statement.
  - Blank lines are ignored.
  - Comment lines are ignored. Comment lines are lines where the first non-blank character is an "\*" or "#".
4. The machine file is composed of blank lines, comment lines, entry lines for the clients, and tag control lines used to decipher the entry lines. The format and content of this file is described further by comments within the BOOTPTAB SAMPLE file.
  5. A machine file (table) must exist for the BOOTPD server to initialize and operate. This file is required because it provides the BootP daemon with information necessary to satisfy client BOOT requests.

However, there are cases when a BootP daemon may not need a *functional* machine file. One example is the case when the BOOTPD server in question acts as a gateway and forwards all requests to another server. In this case, a machine file which contains only blank lines or comment lines may be used for the gateway server.

## BOOTPD Command

6. **STAYUP** is needed only when the TCP/IP machine does not contain an entry for the virtual machine running BOOTPD.

### H.4.4 Messages

Error messages may include:

TCPBOO0001E Machine File was not specified  
TCPBOO0001E Configuration File was not specified  
TCPBOO0004E Unrecognized option *option* - continuing  
TCPBOO0005E Unrecognized subcommand or statement *statement name* - continuing  
TCPBOO0006E Unrecognized operand - *operand*  
TCPBOO0101E RC=*rc* loading Machine file: *fn ft fm*  
TCPBOO0102E RC=*rc* trying to start communications  
TCPBOO0106E Unable to read file: *file*  
TCPBOO0200E Incorrect CMS level. CMS 11 or later is required  
TCPBOO0205E Error encountered while attempting to NUCXLOAD Error encountered while running RXSOCK13, rc=*retcode*  
TCPBOO0400E Unable to determine IP address for *ipaddr*  
TCPBOO0400E Will not answer htype=*htype* chaddr=*chaddr*  
TCPBOO0500E Unrecognized Adapter IP Address: *operand*  
TCPBOO0500E Unrecognized Gateway IP Address: *operand*  
TCPBOO0500E Unrecognized target IP Address: *operand*  
TCPBOO0502E Target IP Address is missing

Information messages may include:

TCPBOO0007I Prior error message refers to line *line* in *file*  
TCPBOO6500I Socket(SendTo) to *port* at *ipaddr* returned: *returned values*  
TCPBOO6501I Socket(Initialize) returned: *returned values*  
TCPBOO6502I Socket(Socket) returned: *returned values*  
TCPBOO6503I Socket(SetSockOpt) returned: *returned values*  
TCPBOO6504I Socket(loctl) returned: *returned values*  
TCPBOO6505I Socket(Bind) returned: *returned values*  
TCPBOO6506I Socket(loCtl,SiocGifConf) returned: *returned values*  
TCPBOO6507I Socket(GetHostId) returned: *returned values*

## BOOTPD Subcommands

TCPBOO6508I Socket(Close) on socket *socket* returned: *returned values*

TCPBOO6509I Socket(loCtl,SiocGifNetMask) returned: *returned values*

TCPBOO6510I Socket(Select) returned: *returned values*

TCPBOO6511I Socket(RecvFrom) returned: *returned values*

### H.5 BOOTPD Subcommands

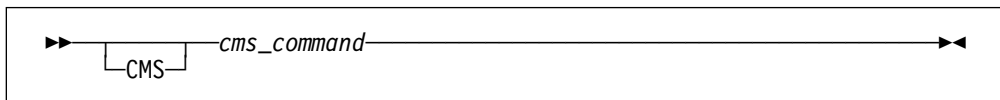
You must be logged on to the BOOTPD server to use the BOOTPD subcommands. The BOOTPD subcommands are listed in Figure 48. This table provides the shortest abbreviation, a description, and a page reference for more information for each BOOTPD subcommand.

Figure 48. BOOTPD Subcommands

Subcommand	Minimum Abbreviation	Description	Page
CMS	CMS	Passes a command to CMS for execution.	210
CONFIG	CONFIG	Displays configuration information.	210
EXCLUDE	EXCLUDE	Identifies adapter addresses for which BOOT requests should be ignored.	214
EXIT	EXIT	Stops the BOOTPD server and its processing. EXIT is equivalent to QUIT and STOP.	215
FORWARD	FORWARD	Controls the forwarding of BOOT requests to another BOOTPD server.	215
HELP	HELP	Displays a summary of BOOTPD subcommands.	219
INCLUDE	INCLUDE	Identifies adapter addresses for which BOOT requests should be handled.	219
LURK	LURK	Toggles the LURK mode of the BOOTPD server.	220
QUIT	QUIT	Stops the BOOTPD server and its processing. QUIT is equivalent to EXIT and STOP.	221
RELOAD	RELOAD	Reloads BOOTPD machine and configuration files.	221
STAYUP	STAYUP	Toggles the STAYUP mode of the BOOTPD server.	223
STOP	STOP	Stops the BOOTPD server and its processing. STOP is equivalent to EXIT and QUIT.	224
TRACE	TRACE	Toggles the TRACE mode of the BOOTPD server.	225

---

## H.6 CMS Subcommand



### H.6.1 Purpose

Use the CMS subcommand to issue a command to CMS.

### H.6.2 Operands

*cms\_command*  
is the CMS command to be issued.

### H.6.3 Usage Notes

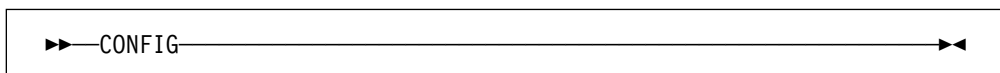
1. Do not issue any CMS command that would take considerable time to execute (for example, XEDIT). While the CMS command executes, the server does not respond to requests.
2. The **CMS** keyword is usually not required because the daemon will pass any command string that is not recognized as a BOOTPD subcommand to CMS. The **CMS** keyword is used to identify commands which would normally be interpreted as a subcommand, for example, **TRACE**.

### H.6.4 Responses

1. After completion of any command, the following ready prompt is displayed:  
BOOTPD Ready;

---

## H.7 CONFIG Subcommand



### H.7.1 Purpose

Use the CONFIG subcommand to display configuration information.



## H.7.2 Usage Notes

1. This subcommand causes the BootP daemon to query the current TCP/IP configuration of the host where the BootP daemon is running, to determine the IP addresses defined for that host. Any included adapter that is not in the defined IP address list will be automatically excluded as the result of this subcommand's operation, whether or not the subcommand completes successfully.

## H.7.3 Responses

1. This subcommand produces a multiple line response which indicates the status of settings, table files used, included and excluded adapter addresses, and forwards that are active or could be activated. This output is discussed below, in sections, for clarity of meaning.

```
Lurk=l Stayup=s Trace=t  
Machine Table=filespec  
Configuration Table=filespec
```

This section indicates the status of the LURK, STAYUP, and TRACE settings, along with the file specifications for the machine and configuration table files used by the server.

*l* is 1 if LURK mode is on; 0 if LURK mode is off.

*s* is 1 if STAYUP mode is on; 0 if STAYUP mode is off.

*t* is 1 if TRACE mode is on; 0 if TRACE mode is off.

*filespec*

is the file name, file type and file mode of the file that is in use.

```
Included Addresses:  
Adapter adpaddr reqtype
```

This section indicates the IP addresses of adapters for which the BootP daemon will process requests. One "Adapter" line is displayed for each adapter that the BootP daemon will handle. If there are no included addresses, "NONE" is displayed instead of the "Adapter" line(s).

*adpaddr*

is the IP address of an adapter on the host system.

*reqtype*

is the type of request that will be handled. Possible values are:

- CLIENTS** for BOOT requests broadcast by clients to the host.
- GATEWAYS** for BOOT requests forwarded by a BootP daemon on behalf of a client.
- ANY** for any client or gateway forwarded requests.

Excluded Addresses:  
Adapter *adpaddr reqtype*

This section indicates the IP addresses of adapters the BootP daemon should ignore. If there are no excluded addresses, "NONE" is displayed instead of the "Adapter" line(s).

*reqtype*

is the type of request that will be excluded. Possible values are:

- CLIENTS** for BOOT requests broadcast by clients to the host.
- GATEWAYS** for BOOT requests forwarded by a BootP daemon on behalf of a client.
- ANY** for any client or gateway forwarded requests.

Forwards:  
Adapter *adpaddr -> toaddr frequency actflag*  
Gateway *gateaddr -> toaddr frequency*

This section indicates whether BOOT request forwarding has been specified for:

- requests received on specific adapters, or
- requests forwarded by a specific gateway.

*adpaddr*

is the IP address of the adapter.

*gateaddr*

is a gateway IP address. The gateway IP address is the address of an adapter on which a request is initially received, but is then forwarded.

*toaddr*

is the IP address of a host that is running another BootP daemon which should receive forwarded requests.

| *frequency*

| is an indication of when forwarding should occur for the specified adapter  
| or gateway. This value can be either:

| **ALWAYS** indicating that any request on the adapter or forwarded by  
| the gateway should always be forwarded.

| **UNKNOWN** indicating that forwarding should occur only for requests on  
| the given adapter, or forwarded by a gateway, when they  
| cannot be handled using this daemon's machine file.

| *actflag*

| indicates the status of the adapter for which the forward has been  
| specified. This value can be either:

| **INCLUDED** indicating that the adapter is included in the configuration  
| and handles both client and gateway-forwarded requests.

| **EXCLUDED** indicating that the adapter is excluded from the  
| configuration (for example, no forwarding will occur until  
| the adapter is included in the configuration).

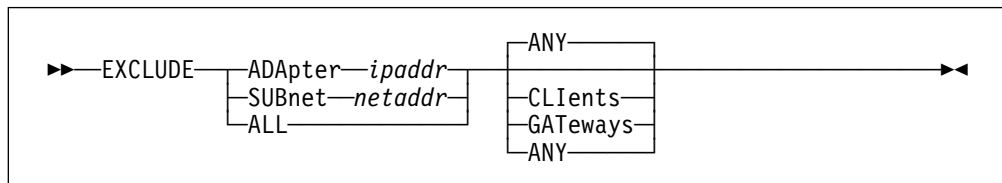
| **PARTIAL** indicating that some BOOT requests received on the  
| adapter will not be handled. This can occur if the  
| **EXCLUDE** or **INCLUDE** subcommand resulted in some  
| BOOT requests not being handled. For example, gateway  
| forwarded requests received over a specific adapter may  
| be excluded, while client requests may be included. For  
| more information about how to control request handling  
| see H.12, "INCLUDE Subcommand" on page 219 and H.8,  
| "EXCLUDE Subcommand" on page 214.

## | H.7.4 Messages

| Error messages may include:

| TCPBOO0003E Too many operands on the subcommand line

## H.8 EXCLUDE Subcommand



### H.8.1 Purpose

Use the EXCLUDE subcommand to specify an adapter address to ignore. You can specify additional operands to indicate whether all requests received across a specific adapter should be ignored, or whether only client BOOT requests or gateway-forwarded requests should be ignored.

### H.8.2 Operands

**ADAPTER** *ipaddr*

indicates the IP address of the adapter on the host system that should be ignored.

**SUBNET** *netaddr*

indicates the address of a subnet that should be ignored. Any host adapter address that is part of the specified subnet is ignored; *netaddr* may be any address valid for the subnet.

**ALL**

indicates that all adapters should be ignored.

**ANY**

indicates that any request that is received on the specified adapters (or adapters associated with a subnet, or "ALL") should be ignored. This is the default.

**CLIENTS**

indicates that only client BOOT requests that are received on the specified adapters (or adapters associated with a subnet, or "ALL") should be ignored.

**GATEWAYS**

indicates that gateway-forwarded requests that are received on the specified adapters (or adapters associated with a subnet, or "ALL") should be ignored.

### H.8.3 Usage Notes

1. The opposite of the EXCLUDE subcommand is the **INCLUDE** subcommand. Any values set by the EXCLUDE subcommand may be reset by the INCLUDE subcommand.
2. This subcommand causes the BootP daemon to query the current TCP/IP configuration of the host where the BootP daemon is running, to determine the IP addresses defined for that host. Any included adapter that is not in the defined IP address list will be automatically excluded as the result of this subcommand's operation, whether or not the subcommand completes successfully.

### H.8.4 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

TCPBOO0006E Unrecognized operand - *operand*

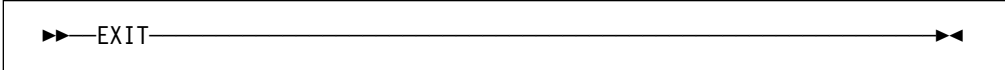
TCPBOO0500E Unrecognized Adapter IP Address: *operand*

TCPBOO0500E Unrecognized Subnet IP Address: *operand*

TCPBOO0500E Unrecognized Gateway IP Address: *operand*

---

## H.9 EXIT Subcommand



### H.9.1 Purpose

Use the EXIT subcommand to stop the BootP daemon. This subcommand is equivalent to the **QUIT** and **STOP** subcommands.

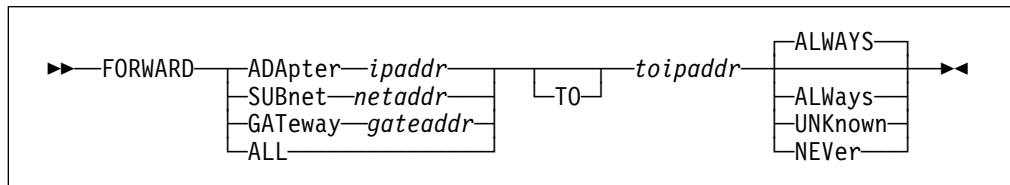
### H.9.2 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

---

## H.10 FORWARD Subcommand



## H.10.1 Purpose

Use the FORWARD subcommand to specify BOOT requests that should be forwarded to another BootP daemon at another IP address. Requests are selected based upon the adapter on which they are received or the gateway from which they were forwarded.

## H.10.2 Operands

### **ADAPTER** *ipaddr*

indicates that BOOT requests received over the specified IP address should be forwarded. The IP address is the address of an adapter on the host system that would receive the request.

### **GATeway** *gateaddr*

indicates that BOOT requests that were forwarded by a gateway at the specified IP address should be forwarded.

### **SUBnet** *netaddr*

indicates that BOOT requests received over the IP addresses that are part of the specified subnet should be forwarded; *netaddr* may be any address valid for the subnet.

### **ALL**

indicates that all IP adapter addresses should be handled as forwarding addresses.

### **TO** *toipaddr*

indicates the IP address to which the BOOT request should be forwarded.

### **ALWAYS**

indicates that BOOT requests that pass the selection criteria (for example, on the specified adapter or from the specified gateway) should always be forwarded. This is the default.

### **NEVER**

indicates that BOOT requests that pass the selection criteria (for example, on the specified adapter or from the specified gateway) should never be forwarded. This cancels the forwarding that may have been previously specified for the BOOT requests that match the criteria.

## UNKNOWN

indicates that BOOT requests that pass the selection criteria (for example, on the specified adapter or from the specified gateway) but cannot be handled using this daemon's machine file should be forwarded. Requests for clients that are in the machine file are not forwarded.

### H.10.3 Usage Notes

1. Forwarding applies only to requests that are not excluded by the **EXCLUDE** subcommand.
2. Forwarding specified for a gateway takes precedence over forwarding specified for an adapter.
3. A hop count is maintained in the BOOT request. This hop count is incremented each time a BootP daemon forwards the request. BOOTPD will not forward a request whose hop count is three or more.
4. The BOOT request contains a server name field. This field allows the client to specify the host name of a BootP daemon which should process the request. If the target BootP daemon is not the receiving server, and the address of the server is not on the same cable as the adapter that received the request, then the request will be forwarded.

Normally, a request is not forwarded to a target BootP daemon when the target daemon is on the same cable as the adapter of the BootP daemon that receives the original request. Such forwarding is not done because it's assumed that the target daemon would have heard this same request.

However, you can override this and force a request to be sent to such a target daemon. If a **FORWARD** to a specific IP address is defined for a receiving adapter, requests will always be forwarded to the BootP daemon at that IP address.

5. Requests that are forwarded by a gateway contain a gateway IP address. BOOTPD specifies this gateway IP address as the address of the adapter which receives the original request. This allows the BootP daemon, which ultimately builds the response packet for the requesting client, to send that packet to the correct gateway; that gateway then sends the response packet to the client.  
  
Only the BootP daemon that initially hears the client request acts as the gateway; subsequent forwarding of the request by other BOOTP daemons does not change the gateway IP address.
6. Forwarding is useful if you wish to centralize your machine files on a specific host and have other BootP daemons forward their received requests to the central site for processing. When you set up forwarding in this manner, you must take into account the increased load on the central server and the time required to forward requests. If the interval to respond to a client request is too

long, that client may then retransmit its requests, and increase the network load.

The following is a simple example of forwarding to central sites. The configuration consists of four hosts that run VM BootP daemons:

<b>VMSAT1, VMSAT2</b>	Satellite VM hosts running BootP daemons connected to subnets with clients that submit BOOT requests.
<b>VMMAIN</b>	Main VM host running BootP for responding to BOOT requests from VMSAT1 and VMSAT2.
<b>VMCENT</b>	Central Master VM HOST containing a master machine file.

In this example, requests received by the VMSAT1 or VMSAT2 are automatically routed to VMMAIN. This allows VMSAT1 and VMSAT2 to run BootP daemons which do not maintain a *functional* machine file. A **FORWARD** statement would appear in the configuration files for VMSAT1 and VMSAT2 as:

```
FORWARD ALL TO xxx.xxx.xxx.xxx ALWAYS
```

where xxx.xxx.xxx.xxx is the IP address of VMMAIN.

Normally, all requests are satisfied by VMMAIN. If VMMAIN cannot handle a request, that request is forwarded to VMCENT, which runs a BootP daemon with the master table file for VMMAIN and other VMMAIN-type hosts. The **FORWARD** statement would appear in the VMMAIN configuration file as:

```
FORWARD ALL TO yyy.yyy.yyy.yyy UNKNOWN
```

where yyy.yyy.yyy.yyy is the IP address of VMCENT.

7. This subcommand causes the BootP daemon to query the current TCP/IP configuration of the host where the BootP daemon is running, to determine the IP addresses defined for that host. Any included adapter that is not in the defined IP address list will be automatically excluded as the result of this subcommand's operation, whether or not the subcommand completes successfully.

## H.10.4 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

TCPBOO0006E Unrecognized operand - *operand*

TCPBOO0500E Unrecognized Adapter IP Address: *operand*

TCPBOO0500E Unrecognized Subnet IP Address: *operand*


TCPBOO0500E Unrecognized Gateway IP Address: *operand*

TCPBOO0500E Unrecognized target IP Address: *operand*



TCPBOO0502E Target IP Address is missing

## H.11 HELP Subcommand



### H.11.1 Purpose

Use the HELP subcommand to display a brief description of available subcommands.

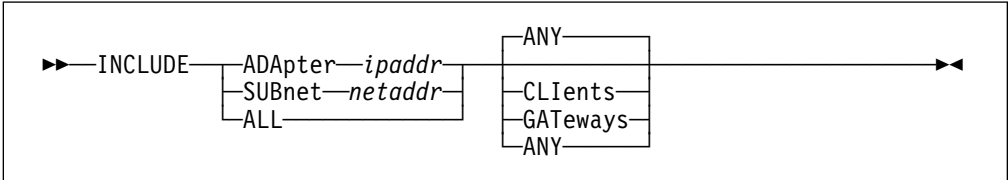
### H.11.2 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

TCPBOO0300 Could not find my help file: *filename* HELP \*

## H.12 INCLUDE Subcommand



### H.12.1 Purpose

Use the INCLUDE subcommand to specify the adapter address or address of BOOTP forwarding daemons for which requests should be handled.

### H.12.2 Operands

**ADAPTER** *ipaddr*

indicates the IP address of the adapter on the host system for which requests should be handled.

**SUBnet** *netaddr*

indicates the address of a subnet for which requests should be handled. Any host adapter address that is part of the specified subnet is handled; *netaddr* may be any address valid for the subnet.

**ALL**

indicates that requests from all adapters should be handled.

|  
| **ANY**

| indicates that any request that is received on the specified adapters (or  
| adapters associated with a subnet, or "ALL") should be handled. This is the  
| default.  
|

| **CLients**

| indicates that only client BOOT requests that are received on the specified  
| adapters (or adapters associated with a subnet, or "ALL") should be handled.  
|

| **GATeways**

| indicates that gateway-forwarded requests that are received on the specified  
| adapters (or adapters associated with a subnet, or "ALL") should be handled.  
|

| **H.12.3 Usage Notes**

- | 1. The opposite of the INCLUDE subcommand is the **EXCLUDE** subcommand.  
| Any values set by the INCLUDE subcommand may be reset by the EXCLUDE  
| subcommand.  
|  
| 2. This subcommand causes the BootP daemon to query the current TCP/IP  
| configuration of the host where the BootP daemon is running, to determine the  
| IP addresses defined for that host. Any included adapter that is not in the  
| defined IP address list will be automatically excluded as the result of this  
| subcommand's operation, whether or not the subcommand completes  
| successfully.  
|

| **H.12.4 Messages**

| Error messages may include:

| TCPBOO0003E Too many operands on the subcommand line

| TCPBOO0006E Unrecognized operand - *operand*


| TCPBOO0500E Unrecognized Adapter IP Address: *operand*

| TCPBOO0500E Unrecognized Subnet IP Address: *operand*

| TCPBOO0500E Unrecognized Gateway IP Address: *operand*  
|

---

| **H.13 LURK Subcommand**  
|

|  **LURK**

### H.13.1 Purpose

Use the LURK subcommand to toggle the LURK mode in the BootP daemon. If the daemon is already operating in LURK mode, it will resume answering requests from clients. If the daemon is not operating in LURK mode, it will begin to listen for, but not will not respond to, client requests.

### H.13.2 Responses

1. The following is displayed upon completion of this subcommand:

LURK is now *l*

where *l* is 0 if LURK mode is off; 1 if LURK mode is on.


### H.13.3 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

---

## H.14 QUIT Subcommand



▶▶ QUIT ◀◀

### H.14.1 Purpose

Use the QUIT subcommand to stop the BootP daemon. This subcommand is equivalent to the **EXIT** and **STOP** subcommands.

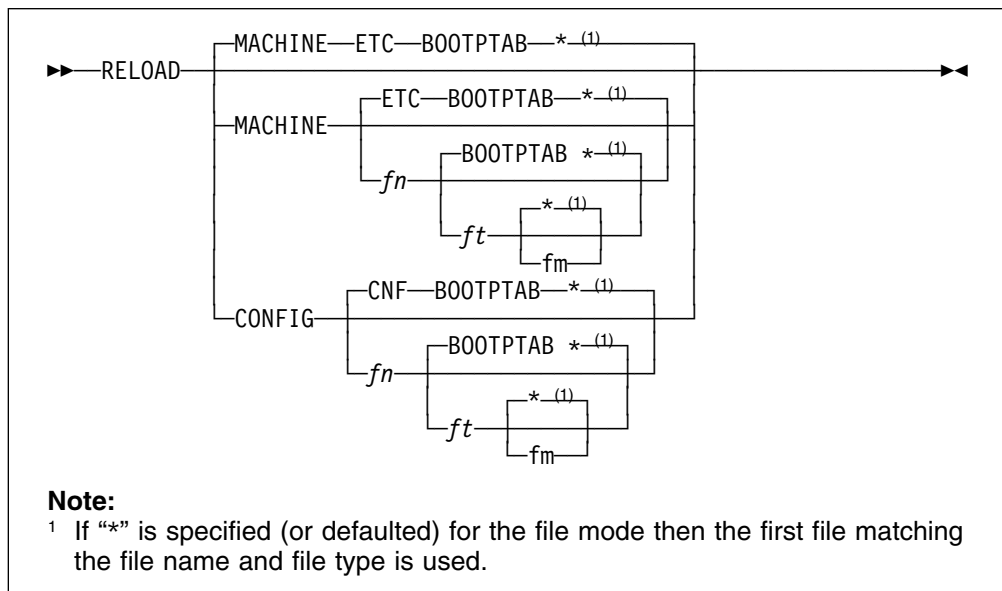
### H.14.2 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

---

## H.15 RELOAD Subcommand



### H.15.1 Purpose

Use the RELOAD subcommand to reload the machine or configuration table file.

### H.15.2 Operands

#### MACHINE

indicates that the file to be loaded is a machine file that contains client information.

*fn* is the file name of the file to load.

*ft* is the file type of the file to load.

*fm* is the file mode of the file to load.

#### CONFIG

indicates that the file to be loaded is a configuration file.

### H.15.3 Usage Notes

1. MACHINE and CONFIG are reserved keywords. They may not be used as file names or file types.
2. The configuration file is composed of blank lines, comment lines, and configuration statements. Configuration statements have the same function and syntax as the BOOTPD **EXCLUDE**, **FORWARD**, and **INCLUDE** subcommands.

The format of the configuration file is:

- One line per statement.
- Blank lines are ignored.
- Comment lines are ignored. Comment lines are lines where the first non-blank character is an "\*" or "#".

3. The machine file is composed of blank lines, comment lines, entry lines for the clients, and tag control lines used to decipher the entry lines. The format and content of this file is described further by comments within the BOOTPTAB SAMPLE file.
4. When a configuration file is processed, all adapters are assumed to be included and no forwarding exists until specified otherwise by configuration file statements.

### H.15.4 Responses

1. If the machine file is successfully loaded, the following response is displayed:

```
fn ft fm loaded in secs seconds.
Table reloaded from fn ft fm
BOOTPD Ready;
```

where *fn*, *ft*, and *fm* are the respective file name, file type and file mode of the loaded file, and *secs* is the time required to load the file.

### H.15.5 Messages

Error messages may include:

```
TCPBOO0003E Too many operands on the subcommand line
TCPBOO0006E Unrecognized operand - operand
TCPBOO0500E Unrecognized Adapter IP Address: operand
TCPBOO0500E Unrecognized Subnet IP Address: operand
TCPBOO0500E Unrecognized Gateway IP Address: operand
TCPBOO0500E Unrecognized target IP Address: operand
TCPBOO0502E Target IP Address is missing
TCPBOO0101E RC=cmd_rc loading Machine file: fn ft fm
```

---

## H.16 STAYUP Subcommand

▶▶ STAYUP ◀◀

### H.16.1 Purpose

Use the STAYUP subcommand to toggle the STAYUP mode in the BOOTP daemon. If the daemon is already operating in STAYUP mode, it will cease operating in this mode and will end processing if a subsequent TCP/IP failure occurs. If the daemon is not operating in STAYUP mode, it will begin to ensure processing will not end if a subsequent TCP/IP failure occurs.

### H.16.2 Responses

1. The following is displayed on completion of the subcommand:

STAYUP is now *s*

where *s* is 0 if STAYUP mode is off; 1 if STAYUP mode is on.

### H.16.3 Usage Notes

1. This subcommand is needed only when the TCP/IP machine does not contain an entry for the virtual machine running BOOTPD.


### H.16.4 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

---

## H.17 STOP Subcommand



### H.17.1 Purpose

Use the STOP subcommand to stop the BootP daemon. This subcommand is equivalent to the **EXIT** and **QUIT** subcommands.


### H.17.2 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

---

## H.18 TRACE Subcommand



▶▶—TRACE—◀◀

### H.18.1 Purpose

Use the TRACE subcommand to toggle the TRACE mode in the BootP daemon. If the daemon is already operating in TRACE mode, it will cease displaying debug information as it processes requests. If the daemon is not operating in TRACE mode, it will display debug information as it processes requests.

### H.18.2 Responses

1. The following is displayed upon completion of this subcommand:

TRACE is now *t*

where *t* is 0 if TRACE mode is off; 1 if TRACE mode is on.

2. See Appendix L, “BOOTPD Trace Records” on page 347 for a description of BOOTPD server trace output.

### H.18.3 Messages

Error messages may include:

TCPBOO0003E Too many operands on the subcommand line

---

## / **Appendix I. Configuring the DHCPD Virtual Machine**

/ The DHCPD virtual machine (daemon) responds to client requests for boot  
/ information using information contained in a DHCP machine file.

/ To configure the DHCPD virtual machine, you must perform the following steps:

### — **DHCPD Configuration Steps** —

1. Update the TCPIP server configuration file.
2. Update the DHCPD profile exit.
3. Configure the ETC DHCPTAB file.

/ This appendix describes the files used to configure DHCPD server, as well as the  
/ DHCPD command and subcommands which control its operation.

---

### / **I.1 Update the TCPIP Server Configuration File**

/ Include DHCPD in the AUTOLOG statement of the TCPIP server configuration file  
/ (PROFILE TCPIP) so that the DHCPD virtual machine is automatically started when  
/ TCPIP is initialized. Verify that the following statements are added to the PROFILE  
/ TCPIP file:

```
AUTOLOG  
  DHCPD 0
```

/ The DHCPD server listens on port 67. Verify that the following statement is added  
/ to your TCPIP server configuration file as well:

```
PORT  
  67 UDP DHCPD ; DHCPD Server
```

/ The DHCPD server uses raw IP functions. Verify that the following statement is  
/ added to your TCPIP server configuration file:

```
OBEY  
  DHCPD  
ENDOBEY
```



---

## / I.2 Update the DHCPD Profile Exit

/ The DHCPD PROFILE EXEC invokes the DHCPDXIT EXEC customization exit (if it  
/ exists) and stacks the TCPRUN EXEC for execution. TCPRUN then starts the  
/ DHCPD server using the TCPRUN global variables OWNER, COMMAND, and  
/ PARMS. You can use the DHCPDXIT customization exit to alter these variables to  
/ change the console log owner, the server module name, or the parameters passed  
/ to the server module. See Chapter 5, “Server Profile Exits” of *TCP/IP V2 R4 for*  
/ *VM: Planning and Customization* for more information about using the  
/ customization exits.

/ **Note:** You must modify the DHCPDXIT EXEC if:

- / • You are using a server module that is not located on the TCP/IP server minidisk  
/ (TCPMAINT 591).
- / • You change parameters passed to the DHCPD command.
- / • You change the user ID of the virtual machine that receives the DHCPD  
/ console output.

/ **Note:** If no parameters are specified in the DHCPDXIT customization exit, DHCPD  
/ is initialized with the following operands:

/ MACHINE ETC DHCPTAB \*

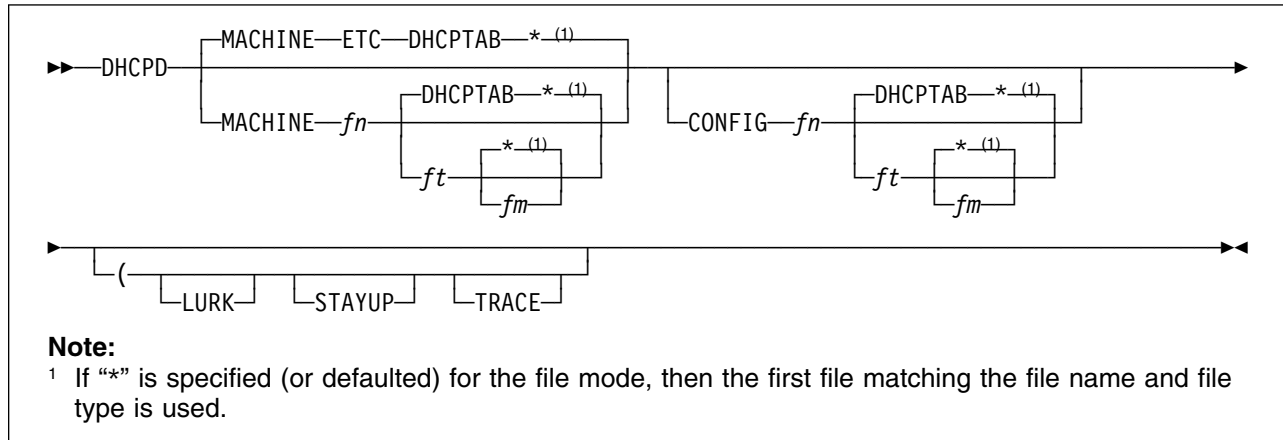
---

## / I.3 Configure the ETC DHCPTAB File

/ The DHCPD server searches the ETC DHCPTAB file for information when it  
/ attempts to satisfy DHCP requests generated by BootP or DHCP clients on the  
/ network. Before you use the DHCPD server, you need to customize the ETC  
/ DHCPTAB file with information about the IBM Network Stations that will be used in  
/ your environment. See I.24, “Constructing a DHCPD Machine File” on page 261  
/ for detailed information about how to specify entries within this file. The customized  
/ ETC DHCPTAB file should remain on the TCPMAINT 198 minidisk.

## DHCPD Command

### I.4 DHCPD Command



#### I.4.1 Purpose

The DHCP daemon (the DHCPD server) responds to client requests for boot information using information contained in a DHCP machine file. This information includes the IP address of the client, the IP address of the TFTP daemon and information about the files to request from the TFTP daemon.

Unlike the BootP daemon, each client (with its assigned IP address) that is serviced by the DHCP daemon does not need to be listed in a machine file. IP addresses may be allocated:

**Dynamically** An IP address is temporarily assigned to the client for a specified period of time. The "lease" on the address must be renewed prior to the end of the specified lease if the client wishes to keep using the IP address.

**Automatically** An IP address is permanently assigned to a client from the list of available addresses.

**Manually** A specific IP address (specified in the machine file) is assigned to the client.

#### I.4.2 Operands

##### **MACHINE**

indicates that the file specification that follows specifies a file containing client information.

*fn* is the file name of the file to load.

*ft* is the file type of the file to load.

/ *fm* is the file mode of the file to load.

/

/ **CONFIG**

/ indicates that the file specification that follows specifies a file containing

/ configuration information. This information lists adapters on the host which

/ should be monitored and those that should not. Also, whether forwarding of

/ BootP/DHCP requests should occur, and when and where they should be sent.

/

/ **LURK**

/ indicates that the DHCP daemon should never respond to a client request. It

/ should only listen.

/

/ **STAYUP**

/ indicates that the DHCP daemon should continue to operate across VM TCP/IP

/ failures.

/

/ **TRACE**

/ indicates that the DHCP daemon should display debug information as requests

/ are processed.

/

### / I.4.3 Usage Notes

/ 1. DHCPD saves information on the addresses that are in use and the clients that

/ are using them in a file on the daemon's A-disk. This file, DHCPD BINDINFO,

/ is read at server initialization, and is updated as the status of supported clients

/ and addresses changes over time.

/

/ A work file, DHCPDWRK BINDINFO, is maintained in addition to the DHCPD

/ BINDINFO file. Neither of these files should be changed by the user.

/

/ 2. MACHINE and CONFIG are reserved keywords. They may not be used as file

/ names or file types.

/

/ 3. The defaults for the DHCPD command when no operands or options are

/ specified are:

/

- / • "ETC DHCPTAB \*" is the machine file,
- / • LURK mode is disabled,
- / • STAYUP mode is disabled,
- / • TRACE mode is disabled,
- / • No configuration file is used; thus, the DHCP daemon will listen for BootP
- / and DHCP requests received on any IP address.

/

/ 4. The configuration file is composed of blank lines, comment lines, and

/ configuration statements. Configuration statements have the same function

/ and syntax as the DHCPD EXCLUDE, FORWARD, and INCLUDE

/ subcommands.

/

/ The format of the configuration file is:

/

- / • One line per statement.
- / • Blank lines are ignored.

/



/ TCPDHC0502E Target IP Address is missing

/

/ Information messages may include:

/ TCPDHC0007I Prior error message refers to line *line* in *file*

/ TCPDHC6500I Socket(SendTo) to *port* at *ipaddr* returned: *returned values*

/ TCPDHC6501I Socket(Initialize) returned: *returned values*

/ TCPDHC6502I Socket(Socket) returned: *returned values*

/ TCPDHC6503I Socket(SetSockOpt) returned: *returned values*

/ TCPDHC6504I Socket(ioctl) returned: *returned values*

/ TCPDHC6505I Socket(Bind) returned: *returned values*

/ TCPDHC6506I Socket(ioctl,SiocGifConf) returned: *returned values*

/ TCPDHC6507I Socket(GetHostId) returned: *returned values*

/ TCPDHC6508I Socket(Close) on socket *socket* returned: *returned values*

/ TCPDHC6509I Socket(ioctl,SiocGifNetMask) returned: *returned values*

/ TCPDHC6510I Socket(Select) returned: *returned values*

/ TCPDHC6511I Socket(RecvFrom) returned: *returned values*

---

### / I.5 DHCPD Subcommands

/ The DHCPD subcommands are listed in Figure 49. This table provides the  
 / shortest abbreviation, a description, and a page reference for more information for  
 / each DHCPD subcommand.

---

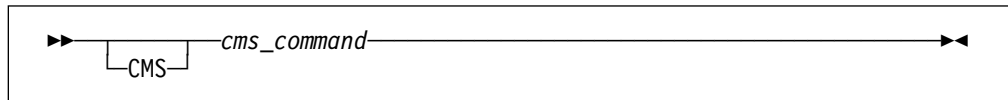
/ *Figure 49 (Page 1 of 2). DHCPD Subcommands*

<b>Subcommand</b>	<b>Minimum Abbreviation</b>	<b>Description</b>	<b>Page</b>
/ CMS	CMS	Passes a command to CMS for execution.	232
/ CONFIG	CONFIG	Displays configuration information.	233
/ DELETE	DELETE	Delete IP address lease allocated to a client.	236
/ EXCLUDE	EXCLUDE	Identifies adapter addresses for which BootP/DHCP requests should be ignored.	238
/ EXIT	EXIT	Stops the DHCPD server and its processing. EXIT is equivalent to QUIT and STOP.	239
/ FORWARD	FORWARD	Controls the forwarding of BootP/DHCP requests to another DHCPD server.	240
/ HELP	HELP	Displays a summary of DHCPD subcommands.	242

/ Figure 49 (Page 2 of 2). DHCPD Subcommands

Subcommand	Minimum Abbreviation	Description	Page
INCLUDE	INCLUDE	Identifies adapter addresses for which BootP/DHCP requests should be handled.	243
LURK	LURK	Toggles the LURK mode of the DHCPD server.	244
QUIT	QUIT	Stops the DHCPD server and its processing. QUIT is equivalent to EXIT and STOP.	245
RELOAD	RELOAD	Reloads DHCPD machine and configuration files.	246
SHOW	SHOW	Displays the information which would be returned to a client that specified parameters similar to the ones specified on the subcommand.	248
STATUS	STATUS	Displays the status of an IP address, Client or Subnet's addresses.	252
STAYUP	STAYUP	Toggles the STAYUP mode of the DHCPD server.	255
STOP	STOP	Stops the DHCPD server and its processing. STOP is equivalent to EXIT and QUIT.	256
TRACE	TRACE	Toggles the TRACE mode of the DHCPD server.	256

## / I.6 CMS Subcommand



### / I.6.1 Purpose

/ Use the CMS subcommand to issue a command to CMS.

### / I.6.2 Operands

/ *cms\_command*  
/ is the CMS command to be issued.

### / I.6.3 Usage Notes

1. Do not issue any CMS command that would take considerable time to execute (for example, XEDIT). While the CMS command executes, the server does not respond to requests.
2. The CMS keyword is usually not required because the daemon will pass any command string that is not recognized as a DHCPD subcommand to CMS.

/ The CMS keyword is used to identify CMS commands which would normally be  
/ interpreted as a DHCPD subcommand, for example, TRACE.

## / I.6.4 Responses

/ 1. After completion of any command, the following ready prompt is displayed:  
/ DHCPD Ready;  
/ or  
/ DHCPD Ready (rc);  
/ if the return code (rc) is not zero.

---

## / I.7 CONFIG Subcommand

/

▶▶—CONFIG—▶▶

### / I.7.1 Purpose

/ Use the CONFIG subcommand to display configuration information.

### / I.7.2 Usage Notes

/ 1. The CONFIG subcommand causes the DHCP daemon to query the current  
/ TCP/IP configuration of the host where the DHCP daemon is running, to  
/ determine the IP addresses defined for that host. Any included adapter that is  
/ not in the defined IP address list will be automatically excluded as the result of  
/ this subcommand's operation, whether the subcommand completes  
/ successfully or not.

### / I.7.3 Responses

/ 1. The CONFIG subcommand produces a multiple line response which indicates  
/ the status of settings, table files used, included and excluded adapter  
/ addresses, and forwards that are active or could be activated. This output is  
/ discussed below, in sections, for clarity of meaning.

/ Lurk=*l* Stayup=*s* Trace=*t*  
/ Machine Table=*filespec*  
/ Configuration Table=*filespec*

/ This section indicates the status of the LURK, STAYUP, and TRACE settings,  
/ along with the file specifications for the machine and configuration table files  
/ used by the server.

/                    *l*    is 1 if LURK mode is on; 0 if LURK mode is off.  
/  
/                    *s*    is 1 if STAYUP mode is on; 0 if STAYUP mode is off.  
/  
/                    *t*    is 1 if TRACE mode is on; 0 if TRACE mode is off.  
/  
/                    *filespec*  
/                    is the file name, file type and file mode of the file that is in use.

Included Addresses:  
Adapter *adpaddr reqtype*

/                    This section indicates the IP addresses of adapters for which the DHCP  
/  
/                    daemon will process requests. One “Adapter” line is displayed for each  
/  
/                    adapter that the DHCP daemon will handle. If there are no included  
/  
/                    addresses, “NONE” is displayed instead of the “Adapter” line(s).

/                    *adpaddr*  
/                    is the IP address of the adapter.

/                    *reqtype*  
/                    is the type of request that will be handled. Possible values are:

- /                    **CLIENTS**            for BootP/DHCP requests broadcast by clients to the host.
- /                    **GATEWAYS**          for BootP/DHCP requests forwarded by a DHCP daemon  
/                    on behalf of a client.
- /                    **ANY**                    for any client or gateway forwarded requests.

Excluded Addresses:  
Adapter *adpaddr reqtype*

/                    This section indicates the IP addresses of adapters the DHCP daemon should  
/  
/                    ignore. If there are no excluded addresses, “NONE” is displayed instead of the  
/  
/                    “Adapter” line(s).

/                    *adpaddr*  
/                    is the IP address of the adapter.

/                    *reqtype*  
/                    is the type of request that will be excluded. Possible values are:

- /                    **CLIENTS**            for BootP/DHCP requests broadcast by clients to the host.
- /                    **GATEWAYS**          for BootP/DHCP requests forwarded by a DHCP daemon  
/                    on behalf of a client.



/ **ANY** for any client or gateway forwarded requests.

```
Forwards:
Adapter adpaddr -> toaddr frequency actflag
Gateway gateaddr -> toaddr frequency
```

/ This section indicates whether BootP/DHCP request forwarding has been  
/ specified for:

- / • requests received on specific adapters, or
- / • requests forwarded by a specific gateway.

/ *adpaddr*  
/ is the IP address of the adapter.

/ *gateaddr*  
/ is a gateway IP address. The gateway IP address is the address of an  
/ adapter on which a request is initially received, but is then forwarded.

/ *toaddr*  
/ is the IP address of a host that is running another DHCP daemon which  
/ should receive forwarded requests.

/ *frequency*  
/ is an indication of when forwarding should occur for the specified adapter  
/ or gateway. ALWAYS indicates that any request on the adapter or  
/ forwarded by the gateway should always be forwarded.

/ *actflag*  
/ indicates the status of the adapter for which the forward has been  
/ specified. This value can be either:

/ **INCLUDED** indicating that the adapter is included in the configuration  
/ and handles both client and gateway-forwarded requests.

/ **EXCLUDED** indicating that the adapter is excluded from the  
/ configuration (for example, no forwarding will occur until  
/ the adapter is included in the configuration).

/ **PARTIAL** indicating that some BootP/DHCP requests received on  
/ the adapter will not be handled. This can occur if the  
/ EXCLUDE or INCLUDE subcommand resulted in some  
/ BootP/DHCP requests not being handled. For example,  
/ gateway forwarded requests received over a specific  
/ adapter may be excluded, while client requests may be  
/ included. For more information about how to control  
/ request handling see I.13, "INCLUDE Subcommand" on

page 243 and I.9, "EXCLUDE Subcommand" on page 238.

## I.7.4 Messages

Error messages may include:

TCPDH0003E Too many operands on the subcommand line

---

## I.8 DELETE Subcommand

```
>> DELETE LEASE [ADDRESS ipaddr] [CLIENT hwtype clientid] <<
```

### I.8.1 Purpose

Use the DELETE subcommand to remove an active lease for an IP address that has been given to a client.

### I.8.2 Operands

#### LEASE

indicates that a lease is being deleted.

#### ADDRESS

indicates that the operand that follows is the IP address whose lease is to be deleted.

#### *ipaddr*

is an IP address specified in dotted-decimal notation.

#### CLIENT

indicates that the operands which follow identify the client that is associated with the IP address whose lease will be deleted.

#### *hwtype*

is the hardware type of the client computer, or 0. The valid client types are defined in STD 2, RFC 1700.

<b>hwtype</b>	<b>Client hardware</b>
---------------	------------------------

<b>1</b>	ethernet ether
----------	----------------

<b>2</b>	ethernet3 ether3
----------	------------------

<b>3</b>	ax.25
----------	-------

<b>4</b>	pronet
----------	--------

<b>5</b>	chaos
----------	-------

```

/          6      token-ring tr
/          7      arcnet
/          8      hyperchannel
/          9      lanstar
/         10      autonet
/         11      localtalk
/         12      localnet
/         13      ultra_link
/         14      smds
/         15      frame_relay
/         16      atm
/         17      ieee802
/         18      fddi

```

*clientID*

is the hexadecimal MAC address or a name which identifies the client. If a name is specified then:

- *hwtype* must be 0
- If the name contains blanks, then it must be enclosed in single or double quotes.

### / **I.8.3 Usage Notes**

1. The DELETE subcommand is useful when you determine that an assigned lease is no longer being used and you wish to make the address available for reassignment. For example, a lease will become available when a BootP client, which has a permanent lease on an address, moves to a different subnet.

### / **I.8.4 Messages**

Error messages may include:

TCPDHC0006E Unrecognized operand - *operand*

TCPDHC0502E Target IP Address is missing

TCPDHC0522E Client ID is missing

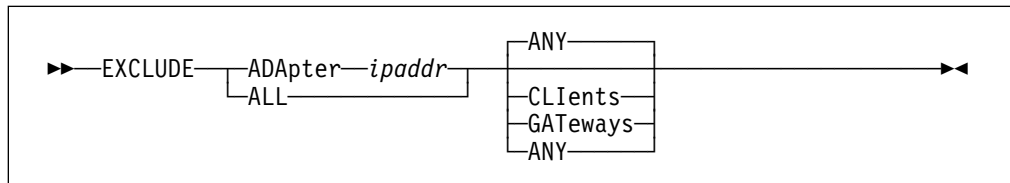
TCPDHC0530E IP address is not valid

TCPDHC0531E Hardware type is not valid

TCPDHC0532E Client ID is not valid

/ TCPDHC0635E The specified client is not associated with an address  
 /  
 / TCPDHC0636E The specified client is associated with more than one address.  
 / You must delete the lease using the address  
 /  
 / TCPDHC0637E *address* is not active

## / I.9 EXCLUDE Subcommand



### / I.9.1 Purpose

/ Use the EXCLUDE subcommand to specify an adapter address to ignore. You can  
 / specify additional operands to indicate whether all requests received across a  
 / specific adapter should be ignored, or whether only client BootP/DHCP requests or  
 / gateway-forwarded requests should be ignored.

### / I.9.2 Operands

/ **ADAPTER** *ipaddr*  
 / indicates the IP address of the adapter on the host system that should be  
 / ignored.

/ **ALL**  
 / indicates that all adapters should be ignored.

/ **ANY**  
 / indicates that any request that is received on the specified adapters (or "ALL")  
 / should be ignored. This is the default.

/ **CLIENTS**  
 / indicates that only client BootP/DHCP requests that are received on the  
 / specified adapters (or "ALL") should be ignored.

/ **GATEWAYS**  
 / indicates that gateway-forwarded requests that are received on the specified  
 / adapters (or "ALL") should be ignored.

### / I.9.3 Usage Notes

- /
1. The opposite of the EXCLUDE subcommand is the INCLUDE subcommand. Any values set by the EXCLUDE subcommand may be reset by the INCLUDE subcommand.
  2. The EXCLUDE subcommand causes the DHCP daemon to query the current TCP/IP configuration of the host where the DHCP daemon is running, to determine the IP addresses defined for that host. Any included adapter that is not in the defined IP address list will be automatically excluded as the result of this subcommand's operation, whether the subcommand completes successfully or not.

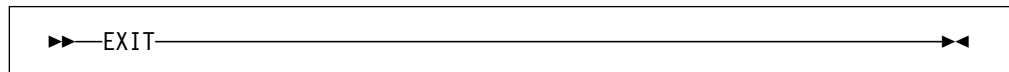
### / I.9.4 Messages

- / Error messages may include:
- / TCPDHC0003E Too many operands on the subcommand line
  - / TCPDHC0006E Unrecognized operand - *operand*
  - / TCPDHC0500E Unrecognized Adapter IP Address: *operand*
  - / TCPDHC0500E Unrecognized Subnet IP Address: *operand*
  - / TCPDHC0500E Unrecognized Gateway IP Address: *operand*

---

## / I.10 EXIT Subcommand

/



### / I.10.1 Purpose

- / Use the EXIT subcommand to stop the DHCP daemon. This subcommand is equivalent to the QUIT and STOP subcommands.

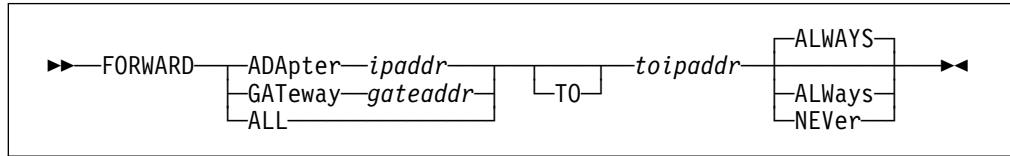
### / I.10.2 Operands

- / The EXIT subcommand has no operands.

### / I.10.3 Messages

- / Error messages may include:
- / TCPDHC0003E Too many operands on the subcommand line

## / I.11 FORWARD Subcommand



### / I.11.1 Purpose

/ Use the FORWARD subcommand to specify BootP/DHCP requests that should be  
/ forwarded to another DHCP daemon at another IP address. Requests are selected  
/ on the basis of the adapter on which they are received or the gateway from which  
/ they were forwarded.

### / I.11.2 Operands

#### / **ADAPTER** *ipaddr*

/ indicates that BootP/DHCP requests received over the specified IP address  
/ should be forwarded. The IP address is the address of an adapter on the host  
/ system that would receive the request.

#### / **GATEWAY** *gateaddr*

/ indicates that BootP/DHCP requests that were forwarded by a gateway at the  
/ specified IP address should be forwarded.

#### / **ALL**

/ indicates that all IP adapter addresses should be handled as forwarding  
/ addresses.

#### / **TO** *toipaddr*

/ indicates the IP address to which the BootP/DHCP request should be  
/ forwarded.

#### / **ALWAYS**

/ indicates that BootP/DHCP requests that pass the selection criteria (for  
/ example, on the specified adapter or from the specified gateway) should always  
/ be forwarded. This is the default.

#### / **NEVER**

/ indicates that BootP/DHCP requests that pass the selection criteria (for  
/ example, on the specified adapter or from the specified gateway) should never  
/ be forwarded. This cancels the forwarding that may have been previously  
/ specified for the BootP/DHCP requests that match the criteria.

### / I.11.3 Usage Notes

- / 1. Forwarding applies only to requests that are not excluded by the EXCLUDE  
/ subcommand.
- / 2. Forwarding specified for a gateway takes precedence over forwarding specified  
/ for an adapter.
- / 3. A hop count is maintained in the BOOT request. This hop count is  
/ incremented each time a DHCP daemon forwards the request. DHCPD will not  
/ forward a request whose hop count is three or more.
- / 4. The BOOT request contains a server name field. This field allows the client to  
/ specify the host name of a DHCP daemon which should process the request.  
/ If the target DHCP daemon is not the receiving server, and the address of the  
/ server is not on the same cable as the adapter that received the request, then  
/ the request will be forwarded.

/ Normally, a request is not forwarded to a target DHCP daemon when the target  
/ daemon is on the same cable as the adapter of the DHCP daemon that  
/ receives the original request. Such forwarding is not done because it is  
/ assumed that the target daemon would have heard this same request.

/ However, you can override this and force a request to be sent to such a target  
/ daemon. If a FORWARD to a specific IP address is defined for a receiving  
/ adapter, requests will always be forwarded to the DHCP daemon at that IP  
/ address.

- / 5. Requests that are forwarded by a gateway contain a gateway IP address.  
/ DHCPD specifies this gateway IP address as the address of the adapter which  
/ receives the original request. This allows the DHCP daemon, which ultimately  
/ builds the response packet for the requesting client, to send that packet to the  
/ correct gateway; that gateway then sends the response packet to the client.

/ Only the DHCP daemon that initially hears the client request acts as the  
/ gateway; subsequent forwarding of the request by other DHCP daemons does  
/ not change the gateway IP address.

- / 6. Forwarding is useful if you wish to centralize your machine files on a specific  
/ host and have other DHCP daemons forward their received requests to the  
/ central site for processing. When you set up forwarding in this manner, you  
/ must take into account the increased load on the central server and the time  
/ required to forward requests. If the interval to respond to a client request is too  
/ long, that client may then retransmit its requests, and increase the network  
/ load.

/ The following is a simple example of forwarding to central sites. The  
/ configuration consists of three hosts that run VM DHCP daemons:

/ **VMSAT1, VMSAT2**            Satellite VM hosts running DHCP daemons  
/                                connected to subnets with clients that submit  
/                                BootP/DHCP requests.

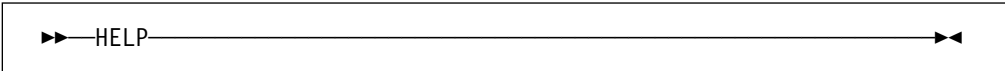
/ **VMMAIN** Main VM host running DHCP for responding to  
 / BootP/DHCP requests from VMSAT1 and VMSAT2.  
 /  
 / In this example, requests received by the VMSAT1 or VMSAT2 are  
 / automatically routed to VMMAIN. This allows VMSAT1 and VMSAT2 to run  
 / DHCP daemons which do not maintain a *functional* machine file. A Forward  
 / statement would appear in the configuration files for VMSAT1 and VMSAT2 as:  
 /  
 / FORWARD ALL TO xxx.xxx.xxx.xxx ALWAYS  
 /  
 / where xxx.xxx.xxx.xxx is the IP address of VMMAIN.  
 /  
 / 7. The FORWARD subcommand causes the DHCP daemon to query the current  
 / TCP/IP configuration of the host where the DHCP daemon is running, to  
 / determine the IP addresses defined for that host. Any included adapter that is  
 / not in the defined IP address list will be automatically excluded as the result of  
 / this subcommand's operation, whether the subcommand completes  
 / successfully or not.

/ **I.11.4 Messages**

/ Error messages may include:  
 /  
 / TCPDHC0003E Too many operands on the subcommand line  
 /  
 / TCPDHC0006E Unrecognized operand - *operand*  
 /  
 / TCPDHC0500E Unrecognized Adapter IP Address: *operand*  
 /  
 / TCPDHC0500E Unrecognized Subnet IP Address: *operand*  
 /  
 / TCPDHC0500E Unrecognized Gateway IP Address: *operand*  
 /  
 / TCPDHC0500E Unrecognized target IP Address: *operand*  
 /  
 / TCPDHC0502E Target IP Address is missing

---

/ **I.12 HELP Subcommand**



/ **I.12.1 Purpose**

/ Use the HELP subcommand to display a brief description of available  
 / subcommands.



## / I.12.2 Operands

/ The HELP subcommand has no operands.

## / I.12.3 Messages

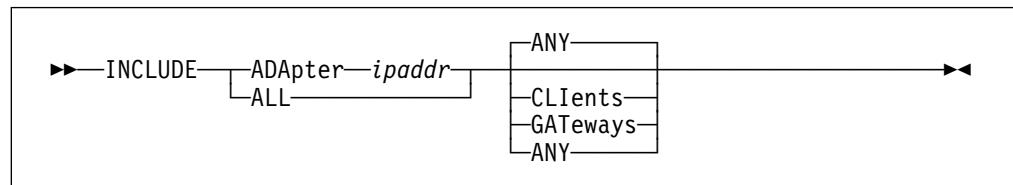
/ Error messages may include:

/ TCPDHC0003E Too many operands on the subcommand line

/ TCPDHC0300 Could not find my help file: *filename* HELP \*

---

## / I.13 INCLUDE Subcommand



### / I.13.1 Purpose

/ Use the INCLUDE subcommand to specify the adapter address or address of  
/ DHCP forwarding daemons for which requests should be handled.

### / I.13.2 Operands

/ **ADAPTER** *ipaddr*

/ indicates the IP address of the adapter on the host system for which requests  
/ should be handled.

/ **ALL**

/ indicates that requests from all adapters should be handled.

/ **ANY**

/ indicates that any request that is received on the specified adapters (or "ALL")  
/ should be handled. This is the default.

/ **CLients**

/ indicates that only client BootP/DHCP requests that are received on the  
/ specified adapters (or "ALL") should be handled.

/ **GATeways**

/ indicates that gateway-forwarded requests that are received on the specified  
/ adapters (or "ALL") should be handled.

### / I.13.3 Usage Notes

- /
1. The opposite of the INCLUDE subcommand is the EXCLUDE subcommand.  
Any values set by the INCLUDE subcommand may be reset by the EXCLUDE subcommand.
  2. The INCLUDE subcommand causes the DHCP daemon to query the current TCP/IP configuration of the host where the DHCP daemon is running, to determine the IP addresses defined for that host. Any included adapter that is not in the defined IP address list will be automatically excluded as the result of this subcommand's operation, whether the subcommand completes successfully or not.

### / I.13.4 Messages

- / Error messages may include:
- / TCPDHC0003E Too many operands on the subcommand line
- / TCPDHC0006E Unrecognized operand - *operand*
- / TCPDHC0500E Unrecognized Adapter IP Address: *operand*
- / TCPDHC0500E Unrecognized Subnet IP Address: *operand*
- / TCPDHC0500E Unrecognized Gateway IP Address: *operand*

---

## / I.14 LURK Subcommand

▶▶—LURK—◀◀

### / I.14.1 Purpose

- / Use the LURK subcommand to toggle the LURK mode in the DHCP daemon. If
- / the daemon is already operating in LURK mode, it will resume answering requests
- / from clients. If the daemon is not operating in LURK mode, it will begin to listen
- / for, but not will not respond to, client requests.

### / I.14.2 Responses

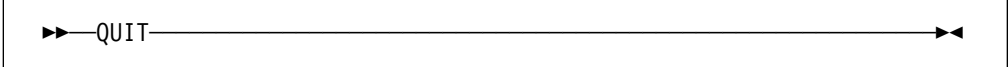
1. The following is displayed on completion of this subcommand:  
LURK is now *l*  
where *l* is 0 if LURK mode is off; 1 if LURK mode is on.

/ **I.14.3 Messages**

/ Error messages may include:  
/ TCPDHC0003E Too many operands on the subcommand line

---

/ **I.15 QUIT Subcommand**

/ 

/ **I.15.1 Purpose**

/ Use the QUIT subcommand to stop the DHCP daemon. This subcommand is  
/ equivalent to the EXIT and STOP subcommands.

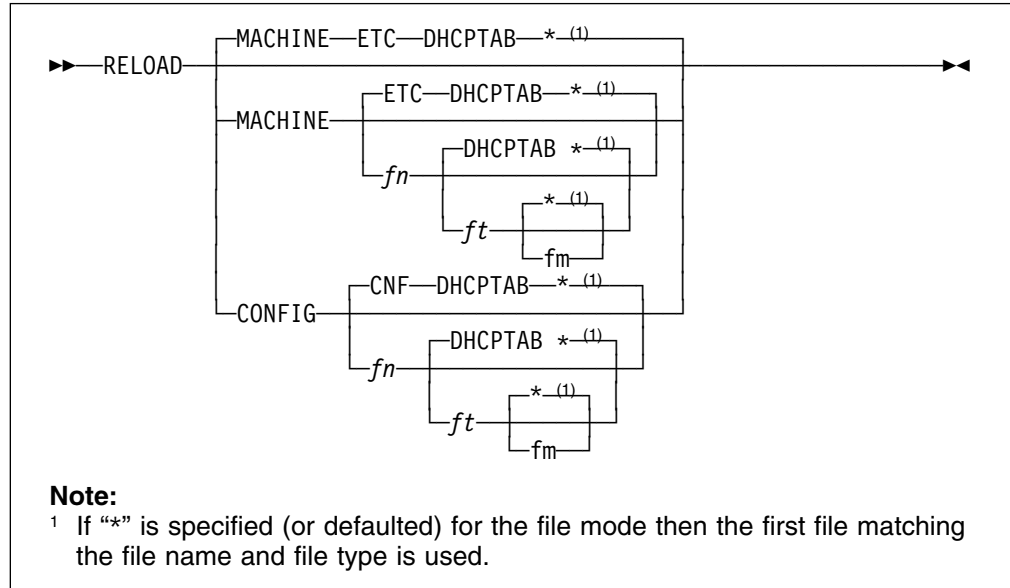
/ **I.15.2 Operands**

/ The QUIT subcommand has no operands.

/ **I.15.3 Messages**

/ Error messages may include:  
/ TCPDHC0003E Too many operands on the subcommand line

## I.16 RELOAD Subcommand



### I.16.1 Purpose

Use the RELOAD subcommand to reload the machine or configuration table file.

### I.16.2 Operands

#### MACHINE

indicates that the file to be loaded is a machine file that contains client information.

*fn* is the file name of the file to load.

*ft* is the file type of the file to load.

*fm* is the file mode of the file to load.

#### CONFIG

indicates that the file to be loaded is a configuration file.

### I.16.3 Usage Notes

1. MACHINE and CONFIG are reserved keywords. They may not be used as file names or file types.
2. The configuration file is composed of blank lines, comment lines, and configuration statements. Configuration statements have the same function

/ and syntax as the DHCPD EXCLUDE, FORWARD, and INCLUDE  
 / subcommands.  
 /  
 / The format of the configuration file is:  
 /  
 / 

- One line per statement.
- Blank lines are ignored.
- Comment lines are ignored. Comment lines are lines where the first  
 / non-blank character is an "\*", "#", or ";".

  
 /  
 / 3. The machine file is composed of blank lines, comment lines, entry lines for the  
 / clients, and tag control lines used to decipher the entry lines. The format and  
 / content of this file is described in I.22, "DHCPD Machine Statements" on  
 / page 257.  
 /  
 / 4. When a configuration file is processed, all adapters are assumed to be included  
 / and no forwarding exists until specified otherwise by configuration file  
 / statements.

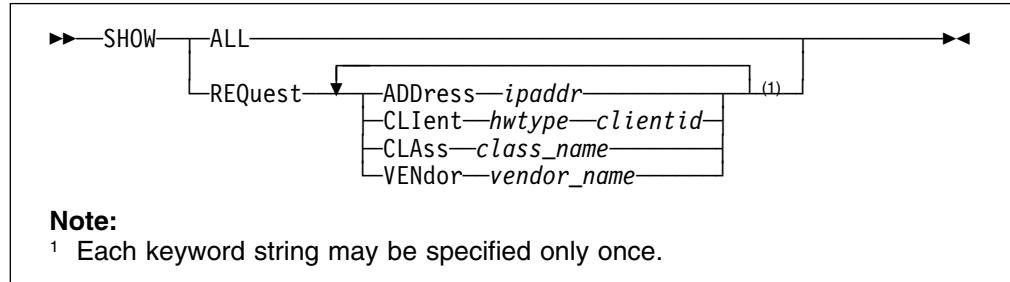
## / I.16.4 Responses

/ 1. If the machine file is successfully loaded, the following response is displayed:  
 /  
 / *fn ft fm* loaded in *secs* seconds.  
 / Table reloaded from *fn ft fm*  
 / DHCPD Ready;  
 /  
 / where *fn*, *ft*, and *fm* are the respective file name, file type and file mode of the  
 / loaded file, and *secs* is the time required to load the file.

## / I.16.5 Messages

/ Error messages may include:  
 /  
 / TCPDHC0003E Too many operands on the subcommand line  
 /  
 / TCPDHC0006E Unrecognized operand - *operand*  
 /  
 / TCPDHC0500E Unrecognized Adapter IP Address: *operand*  
 /  
 / TCPDHC0500E Unrecognized Subnet IP Address: *operand*  
 /  
 / TCPDHC0500E Unrecognized Gateway IP Address: *operand*  
 /  
 / TCPDHC0500E Unrecognized target IP Address: *operand*  
 /  
 / TCPDHC0502E Target IP Address is missing  
 /  
 / TCPDHC0101E RC=*cmd\_rc* loading Machine file: *fn ft fm*

## I.17 SHOW Subcommand



### I.17.1 Purpose

Use the SHOW subcommand to display information which would be returned to a client that specified parameters similar to those specified with the SHOW subcommand.

### I.17.2 Operands

#### ALL

indicates that the current client and configuration settings in the machine file should be displayed.

#### REQuest

indicates that the operands which follow identify the client BootP/DHCP request. The output of this command should show the options and addresses that would be used if a client provided a request specifying specific client related options.

#### ADdRes

indicates that the operand which follows is an IP address from which the request would have been received. This can be the subnet address or a valid address on the subnet.

#### *ipaddr*

is an IP address specified in dotted-decimal notation.

#### CLient

indicates that the two operands which follow specify a client ID.

#### *hwtype*

is the hardware type of the client computer, or 0. The valid client types are defined in STD 2, RFC 1700.

<b>hwtype</b>	<b>Client hardware</b>
---------------	------------------------

<b>1</b>	ethernet ether
----------	----------------

<b>2</b>	ethernet3 ether3
----------	------------------

```

/          3      ax.25
/          4      pronet
/          5      chaos
/          6      token-ring tr
/          7      arcnet
/          8      hyperchannel
/          9      lanstar
/         10      autonet
/         11      localtalk
/         12      localnet
/         13      ultra_link
/         14      smds
/         15      frame_relay
/         16      atm
/         17      ieee802
/         18      fddi

```

*clientid*

is the hexadecimal MAC address or a name which identifies the client. If a name is specified then:

- *hwtype* must be 0
- If the name contains blanks, then it must be enclosed in single or double quotes.

**CLAss**

indicates that the operand which follows specifies a class name.

*class\_name*

is the user-defined label that identifies the class. The client would specify the class name using option 77. The class name is an ASCII string of up to 255 characters (for example, "accounting"). If the class name contains spaces, it must be surrounded by a pair of single quotes (') or a pair of double quotes (").

**VENdor**

indicates that the operand which follows specifies a vendor name.

*vendor\_name*

is the user-defined label that identifies the vendor. The client would transmit this label using option 60. The vendor name is an ASCII string of up to 255

/ characters (for example, "IBM"). If the vendor name contains spaces, it must  
/ be surrounded by a pair of single quotes (') or a pair of double quotes (").

### / I.17.3 Responses

/ 1. If the "ALL" operand is specified, then output similar to the following is  
/ displayed.

```
/ SHOW ALL  
/ GLOBAL DATA  
/   Boot Strap Server: 9.100.48.75  
/   Support BootP: Yes  
/   Support Unlisted Clients: No  
/   Lease Expire Interval: 1 MINUTES  
/   Lease Time Default: -1  
/   Ping Time: 1 SECONDS  
/   Reserved Time: 5 MINUTES  
/   Used IP Expire Interval: 30 SECONDS  
/   OPTION 4: 9.100.48.50  
/   VENDOR: IBM Network Station  
/     OPTION 43: /  
/   CLIENT: 6 0000e580fca8  
/     IP ADDRESS: ANY  
/     OPTION 4: 9.100.48.75  
/   CLASS: IBMNSM 1.0.0  
/     OPTION 67: /QIBM/ProdData/NetworkStation/kernel  
/ SUBNET 9.100.57.0  
/   SUBNET MASK: 255.255.255.0  
/   RANGE: 9.100.57.1 - 9.100.57.99  
/   OPTION 3: 9.100.57.253  
/   OPTION 5: 9.100.25.252  
/   OPTION 6: 9.100.25.252  
/   OPTION 15: ibm.com  
/   CLASS: IBMNSM 1.0.1  
/     OPTION 67: /QIBM/ProdData/NetworkStation/me  
/ SUBNET 9.100.57.0  
/   SUBNET MASK: 255.255.255.0  
/   RANGE: 9.100.57.100 - 9.100.57.140  
/   OPTION 3: 9.100.57.253  
/   OPTION 5: 9.100.25.252  
/   OPTION 6: 9.100.25.252  
/   OPTION 15: ibm.com  
/   CLASS: IBMNSM 1.0.1  
/     OPTION 67: /QIBM/ProdData/NetworkStation/altme
```

/ The Option lines in the output indicate the configuration options that are defined  
/ in the machine file. These options are sent to the client in the DHCP and  
/ BootP replies.



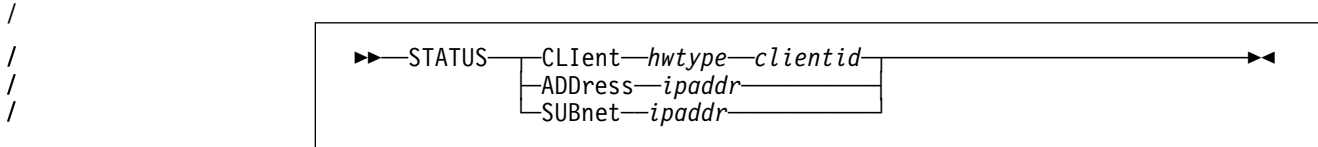


## / I.17.5 Messages

/ Error messages may include:  
/  
/ TCPDHC0003E Too many operands on the subcommand line  
/  
/ TCPDHC0006E Unrecognized operand - *operand*  
/  
/ TCPDHC0502E Target IP address is missing  
/  
/ TCPDHC0522E Client ID is missing  
/  
/ TCPDHC0523E Class name is missing  
/  
/ TCPDHC0524E Vendor name is missing  
/  
/ TCPDHC0530E IP address is not valid  
/  
/ TCPDHC0531E Hardware type is not valid  
/  
/ TCPDHC0532E Client ID is not valid  
/  
/ TCPDHC0624E CLASS name is missing or not valid  
/  
/ TCPDHC0626E VENDOR name is missing or not valid

---

## / I.18 STATUS Subcommand



### / I.18.1 Purpose

/ Use the STATUS subcommand to display the status of an IP address, Client or  
/ Subnet's addresses.

### / I.18.2 Operands

#### / **CLient**

/ indicates that the two operands which follow specify a client ID. The status of  
/ the IP address currently leased to the client should be shown.

#### / *hwtype*

/ is the hardware type of the client computer, or 0. The valid client types are  
/ defined in STD 2, RFC 1700.

#### / **hwtype Client hardware**

/ **1** ethernet ether  
/  
/ **2** ethernet3 ether3  
/  
/ **3** ax.25

/	<b>4</b>	pronet
/	<b>5</b>	chaos
/	<b>6</b>	token-ring tr
/	<b>7</b>	arcnet
/	<b>8</b>	hyperchannel
/	<b>9</b>	lanstar
/	<b>10</b>	autonet
/	<b>11</b>	localtalk
/	<b>12</b>	localnet
/	<b>13</b>	ultra_link
/	<b>14</b>	smds
/	<b>15</b>	frame_relay
/	<b>16</b>	atm
/	<b>17</b>	ieee802
/	<b>18</b>	fddi

*clientid*

is the hexadecimal MAC address or a name which identifies the client. If a name is specified then:

- *hwtype* must be 0
- If the name contains blanks, then it must be enclosed in single or double quotes.

**ADDRESS**

indicates that the operand which follows is an IP address whose status should be shown.

*ipaddr*

is an IP address specified in dotted-decimal notation.

**SUBnet**

indicates that the operand which follows is an IP address of a subnet. The status of the IP addresses associated with the subnet should be shown.

### / I.18.3 Responses

- /
1. If a specific address is being queried by either the CLIENT or ADDRESS operand then the following would be shown.
- /
- If the address is available for reassignment:  
/ STATUS ADDRESS 9.100.57.110  
/ 9.100.57.110 AVAILABLE
  - If the lease for this address has expired, but the address is waiting to re-enter the available pool:  
/ STATUS ADDRESS 9.100.57.111  
/ 9.100.57.111 EXPIRED. HELD UNTIL 23:59:00 ON 24 DECEMBER 1997  
/ TO: 0 BEACH
  - If an address is currently leased to a client:  
/ STATUS ADDRESS 9.100.57.112  
/ 9.100.57.112 IN USE UNTIL 23:59:00 ON 24 DECEMBER 1997  
/ TO: 0 STEVEGESSNER
  - If an address is permanently leased to a client:  
/ STATUS ADDRESS 9.100.57.113  
/ 9.100.57.113 IN USE PERMANENTLY  
/ TO: 0 ADAMGESSNER
  - If an address is being pinged before a DHCP response is sent:  
/ STATUS ADDRESS 9.100.57.114  
/ 9.100.57.114 WAITING FOR ICMP REPLY BEFORE BEING OFFERED  
/ TO: 0 PAULAGESSNER
  - If an address is being offered to a client:  
/ 9.100.57.115 BEING OFFERED  
/ TO: 0 MARK
  - If an address was excluded or not specified in the machine file:  
/ STATUS ADDRESS 9.100.57.100  
/ 9.100.57.110 EXCLUDED
- /
2. If a pool of addresses is being queried, then the output will provide information for each IP address that belongs to the pool, and which is not excluded. In the following example, subnet 9.100.57.0 has 6 addresses defined from 110 to 115.
- /



/ **I.19.2 Responses**

- / 1. The following is displayed on completion of the subcommand:  
/ STAYUP is now s  
/ where s is 0 if STAYUP mode is off; 1 if STAYUP mode is on.

/ **I.19.3 Usage Notes**

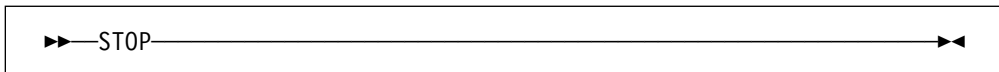
- / 1. The STAYUP subcommand is needed only when the TCP/IP machine does not  
/ contain an entry for the virtual machine running DHCPD.

/ **I.19.4 Messages**

- / Error messages may include:  
/ TCPDHC0003E Too many operands on the subcommand line

---

/ **I.20 STOP Subcommand**



/ **I.20.1 Purpose**

- / Use the STOP subcommand to stop the DHCP daemon. This subcommand is  
/ equivalent to the EXIT and QUIT subcommands.

/ **I.20.2 Operands**

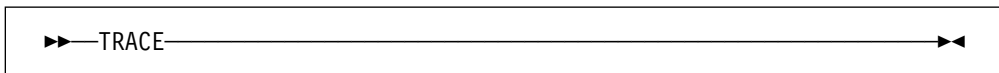
- / The STOP subcommand has no operands.

/ **I.20.3 Messages**

- / Error messages may include:  
/ TCPDHC0003E Too many operands on the subcommand line

---

/ **I.21 TRACE Subcommand**



### / I.21.1 Purpose

/ Use the TRACE subcommand to toggle the TRACE mode in the DHCP daemon. If  
 / the daemon is already operating in TRACE mode, it will cease displaying debug  
 / information as it processes requests. If the daemon is not operating in TRACE  
 / mode, it will display debug information as it processes requests.

### / I.21.2 Responses

/ 1. The following is displayed on completion of this subcommand:  
 / TRACE is now *t*  
 / where *t* is 0 if TRACE mode is off; 1 if TRACE mode is on.  
 / 2. See Appendix M, “DHCPD Trace Records” on page 349 for a description of  
 / DHCPD server trace output.

### / I.21.3 Messages

/ Error messages may include:  
 / TCPDHC0003E Too many operands on the subcommand line

## / I.22 DHCPD Machine Statements

### / I.22.1 Machine File Overview

/ The DHCP server provides machine information to clients based on statements  
 / contained in the server's machine file and based on information provided by the  
 / client. The server's machine file defines the policy for allocating IP addresses and  
 / other configuration parameters. The file is a “map” that the server uses to  
 / determine what information should be provided to the requesting client.

/ Before you start the DHCP server, you must have created the machine file. Once  
 / the DHCP server program is running, you can also make dynamic changes to the  
 / configuration by modifying the machine file and using the RELOAD subcommand to  
 / update the machine information. For more information on the RELOAD  
 / subcommand, see I.16, “RELOAD Subcommand” on page 246.

/ You create a hierarchy of machine parameters for a DHCP-supported network by  
 / specifying some configuration values that are served globally to all clients, while  
 / other configuration values are served only to certain clients. Serving different  
 / configuration information to clients is often based on network location, equipment  
 / vendor, or user characteristics.

/ Depending on your configuration, you can specify subnets, classes, vendors, and  
 / clients to provide configuration information to different groups of clients:

## DHCPD Machine Statements

- When defined globally, client, vendor or class options are available to DHCP clients regardless of their network location.
- Parameters specified for a subnet, class, or client are considered local to the subnet, class, or client. A client defined within a subnet inherits both the global options and the options defined for that subnet. If a parameter is specified in more than one level in the network hierarchy, the lowest level (which is the most specific) is used.
- Use the Subnet statement to specify configuration parameters for one subnet for a specific location in your network or enterprise. The task of configuring subnets also requires you to set lease time and other options for clients using the subnet.
  - Use the Class statement to configure DHCP classes to provide unique configuration information from the server to clients that identify themselves as belonging to that class. For example, a group of clients that all use a shared printer could comprise a Class.
  - Use a Vendor statement to provide unique configuration information to clients that identify themselves as using a specific vendor's equipment or software. Specially-defined options may be served to these clients.
  - Use a Client statement in the DHCP server configuration file to serve specified options to a specific client or to exclude that client from service. You can also use a Client statement to exclude IP addresses from service.
- The concept of scoped statements in DHCP machine file is shown by the following:



## DHCPD Machine Statements

```
/ option A 1
/ option B 2
/ option C 3

/ client k
/ {
/ }

/ subnet y
/ {
/ client m
/ {
/ option E 11
/ option H 12
/ }
/ }

/ class Q
/ {
/ option G
/ option F 9
/ }

/ option C 6
/ option E 5
/ }

/ subnet z
/ {
/ option F 7
/ }

/ class x
/ {
/ option E 1
/ }
```

In this example:

- Options A, B, and C are global and are inherited by all clients in the network unless overridden by a value for the same option at a lower level in the network. Client K is a specific client specified at the global level.
- Clients that are specifically defined under a subnet, such as client M, must be located in that subnet in order to be served. Client M must indicate it is in subnet Y in order to be served.
- Clients that are not specifically defined will automatically fall into a specific location in the hierarchy based on their current network location.
- Option G is served only to clients that belong to class Q in subnet Y.



/

/                   – If a class is specified at both a global level and a subnet level, then the

/                    subnet level class parameters take precedence over the global level class

/                    parameters.

/

- Vendor statements always have a global scope.

/

/                   For example, if option 1 is specified as both a class option at the global level and

/                    as a subnet level option, then the class level option is used because it is more

/                    specific to the client's environment.

---

## / **I.24 Constructing a DHCPD Machine File**

/                   The machine file defines the information that will be returned to clients as

/                    configuration parameters and determines how addresses are to be assigned. It is

/                    important for you to layout the information in the simplest manner possible. This

/                    section will walk you through the construction of a machine file in a step by step

/                    manner. For the sake of a robust example, we will override most of the statement

/                    defaults in order to discuss them and construct a large sample machine file. In

/                    your case, you will probably be able to rely on the defaults.

### / **I.24.1 Global Information**

/                   Certain information is global to all clients that are being serviced or relates to the

/                    overall operation of the server. This information should be determined first and

/                    placed at the global level.

#### / **I.24.1.1 Ping Time**

/                   Prior to assigning an IP address to a client for the first time, DHCPD will issue an

/                    ICMP Echo Request to that address (in other words, PING it). The amount of time

/                    it will wait for a response before assigning that address is determined by the

/                    PingTime statement. If the DHCPD server receives a response from a machine

/                    actively using the address, then it knows that there is a problem and it can not

/                    reissue the address. The default is one second. If more time is needed within your

/                    environment for a ping response to reach your DHCPD server, then add the

/                    PingTime statement.

/                   One way to determine whether your installation will need more time is to use the

/                    TCP/IP PING command to sample the ping response times for the addresses that

/                    will be supported. By choosing addresses of active machines already on the target

/                    subnets you can determine how long it takes them to respond and thereby

/                    determine an appropriate time interval.

/                   For our example, the default time of 1 second is too small so a new PingTime of 2

/                    seconds is set.

/                   PingTime 2 SEC

### **I.24.1.2 Reserved Address Time**

If the client requests boot parameters using DHCP protocol, then it expects that it could receive multiple offers of addresses if there is more than one DHCP server listening. Generally, the client will wait a specified period of time to receive all DHCP offers, then select one. Since there is the possibility that another server might be selected *and* our DHCPD server might not see the accept packet (due to an error somewhere), DHCPD needs to know how long to reserve an IP address before putting it back into the address pool for reuse. The `ReservedTime` statement lets you change the time the server will wait before giving up on an offer and reusing the address. In our case, assume that our clients have been set to respond to an offer in one minute instead of the default of five minutes.

```
ReservedTime 1 minute
```

### **I.24.1.3 Setting the Lease Duration**

DHCPD can reserve addresses either indefinitely or for a particular lease time. Indefinite leases are used primarily for BootP clients, because these clients do not support the concept of leasing an address for a specified period of time. Specific lease intervals are used for most DHCP clients. The clients are provided an address for a specified period of time, and may extend the lease during their lease time.

For our example, the default lease period of 24 hours is too long. We have a limited number of addresses that must be reused as one shift of users leaves for the day, and another takes over, using different hardware. For this reason, we want a lease time default of 9 hours. The `LeaseTimeDefault` statement allows us to set that lease interval.

```
LeaseTimeDefault 9 hours
```

### **I.24.1.4 Setting Wait Time Before Reusing an Expired Lease**

When a lease on an address expires, it is best to let the server wait a period of time before returning the address to the address pool, because it is possible that the client might have been delayed in requesting to extend the lease. The `UsedIPAddressExpireInterval` lets you tailor how long the server will wait before returning an address to the address pool for reuse. For our example, the default of 30 seconds is too short; double it to one minute.

```
UsedIPAddressExpireInterval 1 minute
```

### **I.24.1.5 Setting How Often to Check Addresses**

DHCPD does not maintain individual timers for each IP address that it has allocated. Instead, it periodically checks the status of all addresses that are being leased. This checking covers IP addresses with active leases, reserved leases and IP addresses whose leases have expired and are waiting to return to the address pool. The default interval for checking lease statuses is one minute. For our example, assume that we do not want the inherent overhead of checking leases

/ every minute and instead use 15 minutes as the interval. Since we only check the  
/ leases every 15 minutes, it is possible for a lease to expire at 1 minute into the  
/ interval and we will not handle it until the check that is done 14 minutes later. With  
/ our example server, this is not a problem; we are going to accept a delay in  
/ handling leases and obtain a benefit of less virtual processor usage. The  
/ LeaseExpireInterval statement lets us set this interval.

/ LeaseExpireInterval 15 seconds

#### **I.24.1.6 Supporting BootP Requests**

/ By default, DHCPD supports BootP requests. You may at some time choose that  
/ only DHCP requests are supported, so all addresses are served as leased  
/ addresses instead of allowing BootP clients to obtain indefinite leases on  
/ addresses. The SupportBootP statement lets you control whether BootP requests  
/ are supported. In order to aid in maintenance of our machine file, we will add the  
/ statement so that anyone looking at the file at a later time will easily know that  
/ BootP requests are supported by the server.

/ SupportBootP yes

#### **I.24.1.7 Supporting Requests from Unlisted Clients**

/ By default, DHCPD will support any client which submits a request. You may wish  
/ to require that the client be specified in the machine file in order to be served an  
/ address. In our example, we will specify the SupportUnlistedClients statement with  
/ the “no” operand.

/ SupportUnlistedClients no

#### **I.24.1.8 Specifying the Next Server to Use in the Boot Process**

/ The clients that receive the response from the DHCP server need to know the IP  
/ address of the next server to contact in the boot process. Normally, this is the  
/ address of a TFTP server that would deliver the boot image to the client.

/ Use the BootStrapServer statement to specify the IP address of the next server.  
/ This statement may be specified at the global, class, subnet or client levels. For  
/ our example, all clients are served by the same server, so specify the information  
/ at the global level.

/ BootStrapServer 9.100.40.75

#### **I.24.1.9 Defining New Configuration Options**

/ Over time, it is possible that new configuration options will be defined, and others  
/ considered. To simplify adding support for a client which requires a new option, a  
/ statement to define options is supported. In our example, one of our clients needs  
/ option 181 which is a list of IP addresses for machines which receive datagrams  
/ from the client machine. Since the server does not recognize option 181, we could  
/ code the option as a hexadecimal value,

/ option 181 hex 0964320109643202

/ Or, we could use the DefineOptions statement to define the required option, then  
/ use this newly-defined option later within the machine file.

```
/ DefineOptions  
/ {  
/   option 181 ipalist  
/ }
```

/ The DefineOptions statement must precede the first Option statement. The Option  
/ definition statements are surrounded by a line which contains a Left Brace ({} and a  
/ line containing a Right Brace (}). The braces indicate that the lines which they  
/ surround are part of the DefineOptions level.

### / **1.24.1.10 Global Options**

/ There will be some options that you wish to serve to all clients supported by your  
/ DHCPD server, regardless of the subnet on which they reside. These options are  
/ defined at the global level of the machine file.

/ In our case, all of our clients receive the address of the RFC 868 Time Server and  
/ the time offset for our area. The Option statement lets you specify the option  
/ value.

```
/ #####  
/ # Time Server data: #  
/ # option 2 -> offset of the time server from UTC in seconds #  
/ # option 4 -> IP address of an RFC 868 time server #  
/ #####  
/ option 2 -18000  
/ option 4 9.100.40.75
```

/ All of the subnets that this server supports use the same Name Server, Domain  
/ Name and Domain Name Server, so we want to specify this information at the  
/ global level.

```
/ #####  
/ # Options related to all subnets served by this server. #  
/ # option 5 -> Name Server IP addresses #  
/ # option 6 -> Domain Name Server IP addresses #  
/ # option 15 -> Domain Name #  
/ #####  
/ option 5 9.100.25.252  
/ option 6 9.100.25.252  
/ option 15 endicott.ibm.com
```

### I.24.1.11 Specifying Option 43 Vendor Data

Some clients receive additional option data that is not architected by the DHCP RFCs. Only the manner in constructing the data is architected. This data is transmitted to the client using option 43. The data for the option is in the form of a one byte option number from 1 to 254, followed by a one byte field indicating the length of the data, followed by the data itself.

DHCPD provides the appropriate option 43 data to clients on a vendor-specific basis. This is done by matching option 60 (Vendor Class Identifier) data provided by the client with a vendor string, specified as part of a Vendor statement within the machine file. In our example, we have one type of client which uses vendor data; that client specifies option 60 with the string "IBMSPG 1.0.0."

```
#####  
# Vendor "IBMSPG 1.0.0" returns option 43. #  
#####  
Vendor "IBMSPG 1.0.0"  
{  
    option 1 "DEBUG ON"  
    option 2 hex 05  
}
```

For our example client, we've defined two options to be returned as option 43 data. We specified option 1 as data to be translated to ASCII; option 2 as a hexadecimal value. Note that the normal rules for handling option values do not apply at the vendor level; the data is assumed to be either ASCII or hexadecimal. The Left Brace ( { ) and Right Brace ( } ) statements signify that these options belong to the preceding Vendor statement (at the vendor level). We chose to code the data to be returned using two Option statements, to make this information more readily apparent. We also could have defined this data through the use of a single "option 43" statement. If this was done, we would have used the "hex" operand, and specified both options — and their associated values — as a single hexadecimal value.

### I.24.1.12 Specifying Global Class Data

Some clients specify option 77 on their request. This option specifies an ASCII string which is the name of the client class; DHCPD compares this string to the values specified on Class statements. These statements indicate a level under which additional options may be specified to be returned to the client. In our example, we recognize a single class, "IBM Network Station." This class needs additional information to enable it to configure.

```

/ #####
/ # IBM Network Station manager data: #
/ # option 67 -> Name of the boot file for the client to request#
/ #####
/ class "IBM Network Station"
/ {
/ option 67 /QIBM/ProdData/NetworkStation/kernel
/ }

```

### I.24.1.13 Specifying Global Client Data

You may need to specify options that are unique to a particular client, and do not depend on the subnet to which that client is attached. The Client statement lets you identify such clients, and options that apply to them.

The Client statement that follows identifies a client machine that may be served when it is attached to any subnet supported by this server. The first two operands of the Client statement identify the client hardware type and MAC address; the third operand indicates that this client may be served any available address. The Left ({} and Right braces (}) then create a client level specific to this client; any option data that follows the Left Brace will be served to only the previously identified client. In our example, Option 12 is used to assign the host name of BIGSHOT to the client.

```

/ #####
/ # Globally defined Clients #
/ #####
/ client 6 0000E5E8DC61 any
/ {
/ option 12 BIGSHOT
/ }

```

Our next Client statement identifies a client machine which may be served only when it is attached to the subnet which supports address 9.100.58.240. The first two operands on the Client statement identify the client by specifying a hardware type of 0, and through an ASCII value (STEVE'S MACHINE), a client identifier; this value will be provided by the client (through the use of option 61) when it issues a boot request. The third operand indicates this client may be served only address 9.100.48.240; if this address is not already allocated, it will be reserved for this specific client. As before, the Left ({} and Right braces (}) are used to define client level information that is unique to the preceding client. In this case, Option 12 assigns the host name of STEVIEG to the client. Option 181 was previously defined by the DefineOptions statement and is used here.

```

/ client 0 "STEVE'S MACHINE" 9.100.58.240
/ {
/ option 12 STEVIEG
/ option 181 9.100.57.110
/ }

```



Usually, it is better to specify clients using only their hardware type and machine address, because this allows for improved performance when client information is located.

Specifying the IP address to assign to a client removes some of the flexibility provided by the DHCPD server. The specified address can be used only by the indicated client; no other client may use the address. It is recommended that you let the DHCPD server choose the address from its pool of available addresses, because this allows the possibility of address sharing. For example, one user might use an address and free it up in time for the address to be given to another user.

## I.24.2 Subnet Related Data

Once you have determined and specified global information for the server, you now have to layout the information unique to the various subnets that the server is supporting. The Subnet statements, which we will discuss shortly, identify the subnets and the pools of addresses that they will serve. Again, Left ({} and Right (}) braces will be used to group Option, and other statements, with specific Subnet statements.

Our example DHCPD server will service the following environment: two floors of a building where four subnets are defined. Two of these subnets are directly attached to the machine running the VM system; activity for the other two subnets will be forwarded by BOOTPD relay agents. This example is based on an actual test environment, where only the IP addresses changed.

Figure 50 (Page 1 of 2). DHCPD Machine File - Example Subnet Environment

Subnet Address	Subnet Information
9.100.57.0	<p><b>Subnet Mask:</b> 255.255.255.0</p> <p><b>Routers:</b> 9.100.57.253</p> <p><b>DHCP Controlled Addresses:</b> 9.100.57.43-9.100.57.99, excluding 9.100.57.50, 9.100.57.60 and 9.100.57.85.</p> <p><b>Additional Requirements:</b> None</p>
9.100.58.0	<p><b>Subnet Mask:</b> 255.255.255.0</p> <p><b>Routers:</b> 9.100.58.253</p> <p><b>DHCP Controlled Addresses:</b> 9.100.58.103</p> <p><b>Additional Requirements:</b> A specific machine of hardware type 6, MAC address 0000E5E78650 is never allowed to be attached to this subnet.</p>

Figure 50 (Page 2 of 2). DHCPD Machine File - Example Subnet Environment

Subnet Address	Subnet Information
9.100.48.0	<p><b>Subnet Mask:</b> 255.255.255.0</p> <p><b>Routers:</b> 9.100.48.253</p> <p><b>DHCP Controlled Addresses:</b> 9.100.48.200-9.100.48.245</p> <p><b>Additional Requirements:</b> Machines of a specific class, "IBM Network Station," must be allocated IP addresses between 9.100.48.200 and 9.100.48.230.</p>
9.100.176.0	<p><b>Subnet Mask:</b> 255.255.255.0</p> <p><b>Routers:</b> 9.100.176.1</p> <p><b>DHCP Controlled Addresses:</b> 9.100.176.20-9.100.176.35 9.100.176.100-9.100.176.105</p> <p><b>Additional Requirements:</b> None</p>

For subnet 9.100.57.0, a large range of addresses have been set aside for the DHCP server to administer. Within that range of addresses there are three addresses that are still in use by non-DHCP/BootP protocol machines. We could code 4 sets of subnet statements for

- 9.100.57.43-9.100.57.49
- 9.100.57.51-9.100.57.59
- 9.100.57.61-9.100.57.84
- 9.100.57.86-9.100.57.99.

Instead it will be more straightforward to code a single subnet statement and exclude the addresses which DHCP should not handle. The subnet statement identifies the subnet address, the subnet mask, and the range of addresses that DHCP allocates for that subnet statement. Either within the subnet level or at the global level, Client statements may be specified to further restrict the range. The hardware type of zero and identifier of zero indicate that the client statement is excluding an address.

```

/ #####
/ # Subnet 9.100.57.0 #
/ # option 3 -> Router IP addresses #
/ # option 1 -> subnet mask (This option is generated by the #
/ # SUBNET statement. It should not be specified #
/ # as an option.) #
/ #####
/ Subnet 9.100.57.0 255.255.255.0 9.100.57.43-9.100.57.99
/ {
/ client 0 0 9.100.57.50
/ client 0 0 9.100.57.60
/ client 0 0 9.100.57.85

```

Along with specifying the addresses to allocate and the subnet mask to return to the requesting clients, we need to indicate the address of a router that is used by this subnet. We will also specify a closing Right Brace (}) to indicate the completion of the subnet level.

```

/ option 3 9.100.57.253
/ }

```

For subnet 9.100.58.0, only one address is available for allocation by DHCP. Also, a specific client machine may never be allowed to attach to this subnet.

```

/ #####
/ # Subnet 9.100.58.0 #
/ # option 3 -> Router IP addresses #
/ # option 1 -> subnet mask (This option is generated by the #
/ # SUBNET statement. It should not be specified #
/ # as an option.) #
/ #####
/ Subnet 9.100.58.0 255.255.255.0 9.100.58.103-9.100.58.103
/ {
/ option 3 9.100.58.253
/ client 6 0000E5E78650 none
/ }

```

For subnet 9.100.48.0, 46 IP addresses are available for allocation by the server, but 31 are reserved for a specific class of machine (IBM Network Station). In this case, we only need to specify a Class statement with an IP address range, because for these machines, we previously specified a similar Class statement at the global level; that statement provides related options for all machines of this class.

```

/ #####
/ # Subnet 9.100.48.0 #
/ # option 3 -> Router IP addresses #
/ # option 1 -> subnet mask (This option is generated by the #
/ # SUBNET statement. It should not be specified #
/ # as an option.) #
/ #####
/ Subnet 9.100.48.0 255.255.255.0 9.100.48.200-9.100.48.245
/ {
/ option 3 9.100.48.253
/ class "IBM Network Station" 9.100.48.200-9.100.48.230
/ }

```

For subnet 9.100.176.0, 22 addresses are available in two address ranges. Because the available address ranges are grouped at different ends of the total subnet address range, we will create two groups of subnet statements, instead of a single range with a lot of Client statements to exclude unwanted addresses. We will also code, at the global level, a BALANCE: statement so that DHCPD will attempt to alternately use the two subnet statements to satisfy requests. The Balance: statement contains a label that is specified on both subnet statements, in our case "sub176." Without the balance statement, all addresses would be chosen from the first range of addresses before the second range was used.

Balance: sub176

```

/ #####
/ # Subnet 9.100.176.0, addresses 20-35 #
/ # option 3 -> Router IP addresses #
/ # option 1 -> subnet mask (This option is generated by the #
/ # SUBNET statement. It should not be specified #
/ # as an option.) #
/ #####
/ Subnet 9.100.176.0 255.255.255.0 9.100.176.20-9.100.176.35 label:sub176
/ {
/ option 3 9.100.176.1
/ }

```

```

/ #####
/ # Subnet 9.100.176.0, addresses 200-230 #
/ # option 3 -> Router IP addresses #
/ # option 1 -> subnet mask (This option is generated by the #
/ # SUBNET statement. It should not be specified #
/ # as an option.) #
/ #####
/ Subnet 9.100.176.0 255.255.255.0 9.100.176.200-9.100.176.230 label:sub176
/ {
/ option 3 9.100.176.1
/ }

```

```

/          This completes the machine file. The following example shows the completed file
/          with the Balance: statement moved to the beginning of the file.
/
/          Balance: sub176
/          PingTime 2 SEC
/          ReservedTime 1 minute
/          LeaseTimeDefault 9 hours
/          UsedIPAddressExpireInterval 1 minute
/          LeaseExpireInterval 15 seconds
/          SupportBootP yes
/          SupportUnlistedClients no
/          BootStrapServer 9.100.40.75
/
/          DefineOptions
/          {
/            option 181 ipalist
/          }
/
/          #####
/          # Time Server data:
/          # option 2 -> offset of the time server from UTC in seconds
/          # option 4 -> IP address of an RFC 868 time server
/          #####
/          option 2 -18000
/          option 4 9.100.40.75
/
/          #####
/          # Options related to all subnets served by this server.
/          # option 5 -> Name Server IP addresses
/          # option 6 -> Domain Name Server IP addresses
/          # option 15 -> Domain Name
/          #####
/          option 5 9.100.25.252
/          option 6 9.100.25.252
/          option 15 endicott.ibm.com
/
/          #####
/          # Vendor "IBMSPG 1.0.0" returns option 43.
/          #####
/          Vendor "IBMSPG 1.0.0"
/          {
/            option 1 "DEBUG ON"
/            option 2 hex 05
/          }
/
/          #####
/          # IBM Network Station manager data:
/          # option 67 -> Name of the boot file for the client to request#
/          #####

```

```

/          class "IBM Network Station"
/          {
/            option 67 /QIBM/ProdData/NetworkStation/kernel
/          }

/          #####
/          # Globally defined Clients
/          #####
/          client 6 0000E5E8DC61 any
/          {
/            option 12 BIGSHOT
/          }

/          client 0 "STEVE'S MACHINE" 9.100.58.240
/          {
/            option 12 STEVIEG
/            option 181 9.100.57.110
/          }

/          #####
/          # Subnet 9.100.57.0
/          # option 3 -> Router IP addresses
/          # option 1 -> subnet mask (This option is generated by the
/          # SUBNET statement. It should not be specified
/          # as an option.)
/          #####
/          Subnet 9.100.57.0 255.255.255.0 9.100.57.43-9.100.57.99
/          {
/            client 0 0 9.100.57.50
/            client 0 0 9.100.57.60
/            client 0 0 9.100.57.85
/            option 3 9.100.57.253
/          }

/          #####
/          # Subnet 9.100.58.0
/          # option 3 -> Router IP addresses
/          # option 1 -> subnet mask (This option is generated by the
/          # SUBNET statement. It should not be specified
/          # as an option.)
/          #####
/          Subnet 9.100.58.0 255.255.255.0 9.100.58.103-9.100.58.103
/          {
/            option 3 9.100.58.253
/            client 6 0000E5E78650 none
/          }

/          #####
/          # Subnet 9.100.48.0
/          #

```

```

/          # option 3 -> Router IP addresses          #
/          # option 1 -> subnet mask (This option is generated by the #
/          #          SUBNET statement. It should not be specified #
/          #          as an option.)                  #
/          #####
/          Subnet 9.100.48.0 255.255.255.0 9.100.48.200-9.100.48.245
/          {
/          option 3 9.100.48.253
/          class "IBM Network Station" 9.100.48.200-9.100.48.230
/          }
/
/          #####
/          # Subnet 9.100.176.0, addresses 20-35          #
/          # option 3 -> Router IP addresses          #
/          # option 1 -> subnet mask (This option is generated by the #
/          #          SUBNET statement. It should not be specified #
/          #          as an option.)                  #
/          #          #####
/          Subnet 9.100.176.0 255.255.255.0 9.100.176.20-9.100.176.35 label:sub176
/          {
/          option 3 9.100.176.1
/          }
/
/          #####
/          # Subnet 9.100.176.0, addresses 200-230      #
/          # option 3 -> Router IP addresses          #
/          # option 1 -> subnet mask (This option is generated by the #
/          #          SUBNET statement. It should not be specified #
/          #          as an option.)                  #
/          #          #####
/          Subnet 9.100.176.0 255.255.255.0 9.100.176.200-9.100.176.230 label:sub176
/          {
/          option 3 9.100.176.1
/          }

```

---

## / I.25 Machine File Format

/ In the DHCP machine file:

- / • Comments must begin with a pound sign (#), semi-colon (;) or asterisk (\*) in column 1.
- / • Text strings that include spaces must be surrounded by single (') or double quotes (").
- / • Parameters to the right of a left parenthesis are ignored. They are supported for compatibility with other IBM DHCP servers.
- / • A continuation character (\) at the end of a line indicates the information is continued on the next line.

- Braces are used to specify statements that are scoped within other statements.
- Class statements may appear at the global level or scoped within a subnet.
- Client statements may appear at the global level or scoped within a subnet.
- Subnet statements may appear only at the global level.
- Keywords are not case sensitive. Capitalization patterns used in this documentation are not required in the configuration file.

Figure 51 (Page 1 of 2). DHCPD Machine File Statements

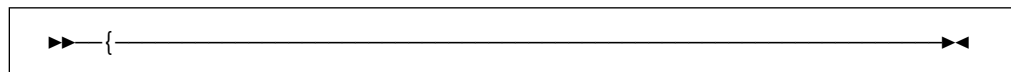
Statement	Minimum Abbreviation	Description	Page
{	{	Indicates the start of a level. Statements which follow the Left Brace are related to the previous statement.	275
}	}	Indicates the end of a level.	276
Balance:	BALANCE:	Specifies subnet groups from which addresses should be assigned in an alternating manner.	276
BootStrapServer	BSS	Specifies the IP address of the next server to contact in the boot strap process. Normally, this will be the address of the TFTP server which delivers the boot image.	277
Class	CLASS	Identifies a user defined class (specified by the client using option 77) and a range of addresses to assign to the class.	278
Client	CLIENT	Identifies a client by an ID, which may be followed by additional options. Can also be used to identify an IP address that should not be used within a range of addresses.	279
DefineOptions	DEFOPTS	Defines (or redefines) the values allowed on Option statements.	282
InOrder:	INORDER:	Specifies which subnet groups should be processed in the order of priority.	281
LeaseExpireInterval	LEI	Specifies the interval at which the lease condition of all addresses in the address pool is examined.	282
LeaseTimeDefault	LTD	Specifies the default lease duration for the leases issued by this server.	283



/ Figure 51 (Page 2 of 2). DHCPD Machine File Statements

Statement	Minimum Abbreviation	Description	Page
Option	OPTION	Defines options to be passed to the client. Also, within the DefineOptions level, you can define (or redefine) how the option values are handled.	284
PingTime	PT	Specifies a time interval that the server will wait for a response to a ping (ICMP Echo Request).	288
ReservedTime	RT	Specifies the maximum amount of time the server holds an offered address in reserve while waiting for a response from the client.	288
Subnet	SUBNET	Specifies configuration parameters for an address pool administered by a server. An address pool is a range of IP addresses to be leased to clients.	289
SupportBootP	SB	Specifies whether DHCPD responds to requests from BootP clients.	292
SupportUnlistedClients	SUC	Controls whether DHCPD will respond to clients that are not listed (by client ID) in the machine file.	292
UsedIPAddressExpireInterval	UIPEI	Specifies the amount of time that an IP address, whose lease has expired, will be held before being returned to the available address pool for possible reassignment to another client.	293
Vendor	VENDOR	Specifies an option 43 value to be returned to clients that specify an option 60 value that matches the vendor name.	294

/ **I.26 { (Left Brace) Statement**



### / I.26.1 Purpose

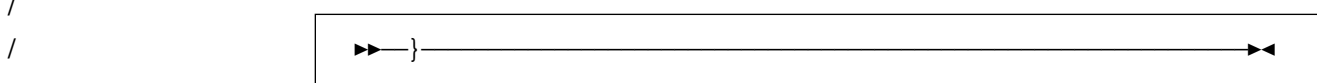
/ Use the Left Brace statement to indicate the start of a level. Statements which  
/ follow the Left Brace are related to the previous Vendor, Subnet, Class, Client or  
/ DefineOptions statement.

### / I.26.2 Usage Notes

/ 1. The Left Brace must be the first non-comment statement in the file or the first  
/ non-comment statement to follow a Vendor, Subnet, Class, Client or  
/ DefineOptions statement.

---

## / I.27 } (Right Brace) Statement

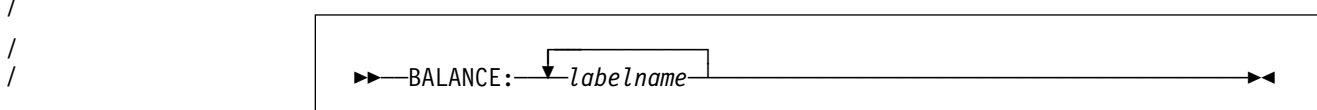


### / I.27.1 Purpose

/ Use the Right Brace statement to indicate the end of a level.

---

## / I.28 Balance: Statement



### / I.28.1 Purpose

/ Use the Balance: statement to specify subnet groups from which addresses should  
/ be assigned in an alternating manner.

### / I.28.2 Operands

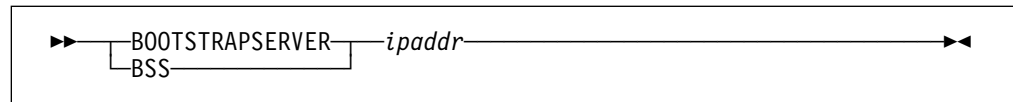
/ *labelname*  
/ is a label (a string of 1 to 64 alphanumeric characters) that identifies the subnet  
/ group.

### / I.28.3 Usage Notes

/ 1. The Balance: statement should be specified at the global level only.  
/ 2. When the Balance: statement is used, DHCP provides the first IP address from  
/ the subnet that is first in the priority list, and subsequent IP addresses from  
/ each lesser-priority subnet, repeating the cycle until addresses are exhausted  
/ equally from all subnets.

/ The following is an example of Balance: processing of a subnet group. IP  
 / addresses are exhausted equally in WIRE1/3 and WIRE1/5: The priority values  
 / of “3” and “5” need not be consecutive. There is no other subnet statement for  
 / WIRE1 with a priority of 1, 2 or 4.  
 /  
 / balance: WIRE1  
 / subnet 9.67.49.0 255.255.255.0 9.67.49.1:9.67.49.100 label:WIRE1/3  
 / subnet 9.67.48.0 255.255.255.0 9.67.48.1:9.67.48.50 label:WIRE1/5  
 /  
 / 3. If a label for a subnet group is specified on Subnet statements but not on a  
 / Balance: or InOrder: statement, then the label on the subnet has no effect.  
 / Clients are allocated addresses for the subnet on which they broadcast the  
 / request.  
 /  
 / For more information on the format of the Subnet statement, see I.39, “SUBNET  
 / Statement” on page 289.

## / I.29 BootStrapServer Statement



### / I.29.1 Purpose

/ Use the BootStrapServer statement to specify the IP address of the next server to  
 / contact in the boot strap process. Normally, this is the address of the TFTP server  
 / from which the client will obtain the boot image after it completes the BootP/DHCP  
 / process of acquiring an IP address and basic configuration information.

### / I.29.2 Operands

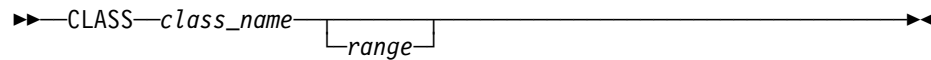
/ *ipaddr*  
 / is an IP address, specified in dotted-decimal notation, which specifies the  
 / address of the next server to contact after completion of the BootP/DHCP  
 / protocol step.

### / I.29.3 Usage Notes

1. The BootStrapServer statement may be specified at the global, subnet, class, and client levels only.
2. The Boot Strap Server address is returned to the client in the “siaddr” field of BootP and DHCP replies. If the BootStrapServer is not specified in the machine table for the selected response, then the primary host IP address for the host running the DHCPD server is used.

---

## / I.30 CLASS Statement



### / I.30.1 Purpose

/ Use the Class statement to identify a user-defined group of clients.

### / I.30.2 Operands

/ *class\_name*

/ is the user-defined label that identifies the class. The client specifies the class name using option 77. The class name is an ASCII string of up to 255 characters (for example, "accounting"). If the class name contains spaces, it must be surrounded by single (') or double (") quotes.

/ *range*

/ is a range of addresses. Enter addresses in dotted-decimal notation, beginning with the lower end of the range, followed by a hyphen, then the upper end of the range, with no spaces between. For example, 9.17.32.1-9.17.32.128.

/ At a global level, a class cannot have a range. A range is only allowed when a class is defined within a subnet. By default, no range is associated with the class. The range must be a subset of the subnet range.

### / I.30.3 Usage Notes

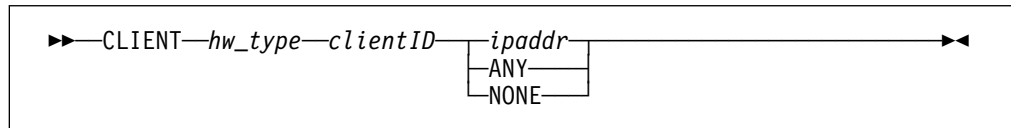
- / 1. The Class statement should be specified at the global level or subnet level.
- / 2. When a Class statement is specified at a Subnet level, the CLASS (and related options) will be used only if:
  - / • The client specified the class with option 77
  - / • The client is on the subnet to which the class belongs
  - / • When an IP address range is specified on the Class statement, then an IP address must be available in the range in order for the Class statement to be used.

/ If an IP address has already been indicated on a matching Client statement, that address must be used and must be within the range of addresses listed on the Class statement.

- / 3. IP addresses specified on the *range* operand are removed from the general pool of addresses used for the subnet. They will be allocated only to clients that belong to the class. This range can be further restricted by Client statements which specify addresses within the CLASS range.

- /
- /
- /
- /
- /
- /
- 4. A client that requests an IP address from a class which has exhausted its range is offered an IP address from the subnet range, if available. The client is not offered the options associated with the exhausted class.
- /
- 5. To assign configuration parameters such as a lease time for all clients in a class, follow the Class statement with Option statements surrounded by braces.
- /

### / I.31 CLIENT Statement



#### / I.31.1 Purpose

/ Use the Client statement to identify a client by an ID, or identify an IP address that should be excluded from a range of available addresses.

#### / I.31.2 Operands

/ *hw\_type*  
/ is the hardware type of the client computer, or 0. The valid client types are defined in STD 2, RFC 1700.

<b>hw_type</b>	<b>Client hardware</b>
1	ethernet ether
2	ethernet3 ether3
3	ax.25
4	pronet
5	chaos
6	token-ring tr
7	arcnet
8	hyperchannel
9	lanstar
10	autonet
11	localtalk
12	localnet
13	ultra_link
14	smds

```

/          15      frame_relay
/
/          16      atm
/
/          17      ieee802
/
/          18      fddi
/
/      clientID
/          is the hexadecimal MAC address or a name which identifies the client. If a
/          name is specified then:
/
/          • hwtype must be 0
/
/          • If the name contains blanks, then it must be enclosed in single or double
/            quotes.
/
/      ipaddr
/          is an IP address, specified in dotted-decimal notation.
/
/      ANY
/          indicates that the server may choose an address from its pool of available
/          addresses.
/
/      NONE
/          indicates that no response is to be returned to client BootP/DHCP requests.

```

### / **I.31.3 Usage Notes**

```

/      1. The Client statement may be specified at the global level or subnet level only.
/
/      2. If the Option statement identifies a client (instead of excluding an address),
/      then it may be followed by Left Brace ({} and Right Brace (}) statements which
/      identify additional statements associated with the Client statement. Option
/      statements are specified within these enclosing braces to associate options
/      with the client. For example:
/
/      client 6 0000E5E8DC60 any
/      {
/      option 12 ADAM
/      }
/
/      3. If the Client statement is being used to restrict an IP address from the list of
/      available addresses, both the hwtype and clientid should be 0. For example:
/
/      client 0 0 9.100.25.110
/
/      While the Client statement can be used to exclude an IP address, a better
/      solution for excluding many IP addresses is to specify a subnet with a range of
/      IP addresses that are never served to clients. For example, the following
/      Subnet statement does not include the first nine addresses on the subnet.
/      These nine addresses are not to be used.
/
/      subnet 9.100.25.0 255.255.255.0 9.100.25.10-9.100.25.254

```

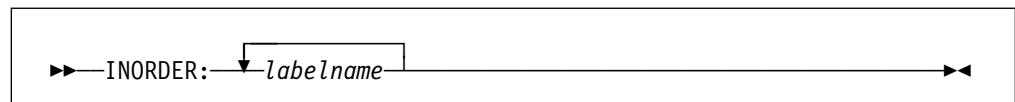
- /
- /
- /
- /
- /
- /
- /
- /
- /
4. The Client statement may be specified at the global level to exclude the client from service by the server, or at a subnet level to exclude a client from service on a particular subnet (address pool). For example:
- client 6 10005aa4b9ab none
5. An IP address restricted to a specific client through the use of the *ipaddr* operand is removed from the general pool of addresses used for a subnet. It will be allocated only to the specified client.

---

/

**I.32 InOrder: Statement**

/



/

**I.32.1 Purpose**

/

Use the `InOrder:` statement to specify subnet groups from which addresses should be assigned based on subnet priorities.

/

**I.32.2 Operands**

/

*labelname*  
is the *labelname* associated with a subnet group.

/

**I.32.3 Usage Notes**

/

1. The `InOrder:` statement should be specified at the global level only.
2. When the `InOrder:` statement is used, the subnets that belong to a listed group are processed in the order established by the priority assigned to each subnet. The subnet address pool with the highest priority within a group is completely exhausted before the subnet address pool with the next highest priority is used. For more information on the `Subnet` statement, see I.39, “`SUBNET` Statement” on page 289.

The following is an example of `InOrder:` processing of subnet group WIRE3. Requests for subnet group WIRE3 exhaust addresses in subnet 9.67.50.0 (WIRE3/1) first, followed by subnet 9.67.51.0 (WIRE3/2), and then 9.67.50.0 (WIRE3/3), which has the same subnet address as WIRE3/1, but specifies a higher address range:

```
InOrder: WIRE3
subnet 9.67.51.0 255.255.255.0 9.67.51.1-9.67.51.50 label:WIRE3/2
subnet 9.67.50.0 255.255.255.0 9.67.50.1-9.67.50.50 label:WIRE3/1
subnet 9.67.50.0 255.255.255.0 9.67.50.51-9.67.50.100 label:WIRE3/3
```

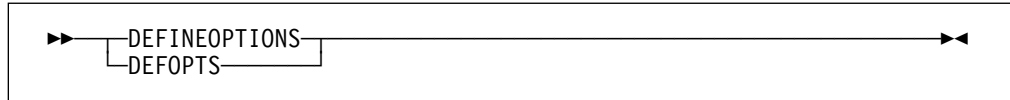
```

/
/      3. If a label for a subnet group is specified on Subnet statements but not on a
/      Balance: or InOrder: statement, then the label on the subnet has no effect.
/      Clients are allocated addresses for the subnet on which they broadcast the
/      request.

```

---

### / I.33 DefineOptions Statement



#### / I.33.1 Purpose

```

/      Use the DefineOptions statement to identify a group of options whose processing is
/      being redefined. This allows you to define how the value specified for an option is
/      handled.

```

#### / I.33.2 Usage Notes

- ```

/      1. The DefineOptions statement should be specified at the global level only and
/      must precede all Option statements.
/
/      2. It may be easier to use the DefineOptions statement to define an unrecognized
/      option than coding the value in ASCII or hexadecimal. For example, if a new
/      option (for example 190) is used by the client to contain a series of 32 bit
/      signed time offsets: 18000, 12000, and 3000, you could code it in hex as
/      follows:

```

```

/      190 hex 0000465000002ee000000bb8

```

```

/      or you could define the options and specify the 190 option values as decimal
/      numbers.

```

```

/      DefineOptions
/      {
/      option 190 lsint(4,*,*)
/      }

```

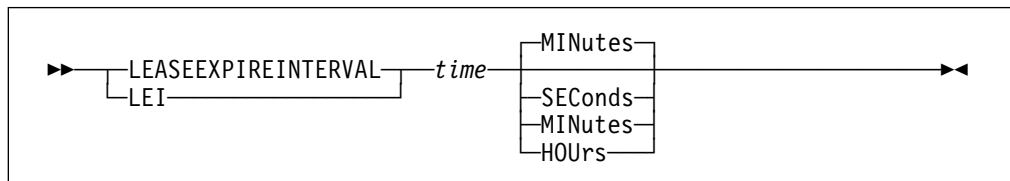
```

/      option 190 18000 12000 3000

```

---

### / I.34 LeaseExpireInterval Statement





/ **I.34.1 Purpose**

/ Use the LeaseExpireInterval statement to specify the interval at which the lease  
/ condition of all addresses in the address pool is examined.

/ **I.34.2 Operands**

/ *time*

/ is the amount of time to wait between checking all addresses for expired  
/ leases.

/ The value must be a positive whole number (Base 10) or “-1” (which means  
/ infinity). If this statement is not specified, the LeaseExpireInterval defaults to 1  
/ minute. The minimum time value must be equal to at least 15 seconds. The  
/ maximum time value must be no more than 12 hours.

/ **SECONDS**

/ indicates the time value is in units of seconds.

/ **MINutes**

/ indicates the time value is in units of minutes.

/ **HOURS**

/ indicates the time value is in units of hours.

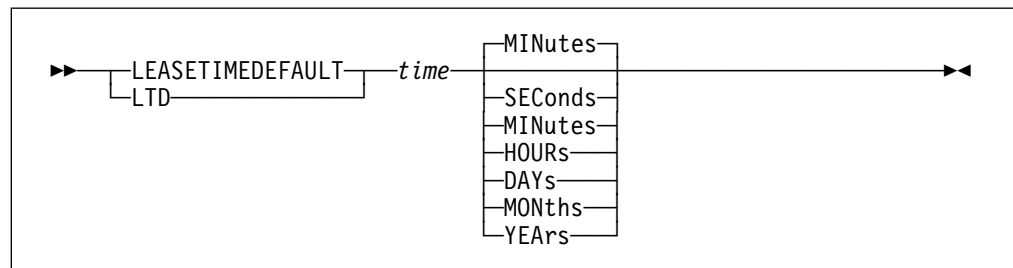
/ **I.34.3 Usage Notes**

/ 1. The LeaseExpireInterval statement should be specified at the global level only  
/ and therefore, is used for all addresses supported by this server.

/ 2. The value specified should be less than the value for LeaseTimeDefault to  
/ ensure that expired leases are returned to the pool in a timely manner.

---

/ **I.35 LeaseTimeDefault Statement**



### / I.35.1 Purpose

/ Use the LeaseTimeDefault statement to specify the default lease duration for the  
/ leases issued by this server.

### / I.35.2 Operands

/ *time*  
/ is the default lease duration.  
/ The value must be a positive whole number (Base 10) or “-1” (which means  
/ infinity). If this statement is not specified, the LeaseTimeDefault defaults to 24  
/ hours. The minimum time value must be equal to at least 180 seconds.

/ **SEConds**  
/ indicates the time value is in units of seconds.

/ **MINutes**  
/ indicates the time value is in units of minutes.

/ **HOUrs**  
/ indicates the time value is in units of hours.

/ **DAYs**  
/ indicates the time value is in units of days.

/ **MONths**  
/ indicates the time value is in units of months.

/ **YEArS**  
/ indicates the time value is in units of years.

### / I.35.3 Usage Notes

- / 1. The LeaseTimeDefault statement should be specified at the global level only  
/ and therefore, is used for all addresses supported by this server.  
/ 2. The LeaseTimeDefault may be overridden within the machine file using option  
/ 51.

---

## / I.36 OPTION Statement

/

## / I.36.1 Purpose

/ Use the Option statement to define options to be passed to the client. Also, within  
/ the DefineOptions level, you can define (or redefine) option values.

/ **Note:** See I.44, “DHCP Options” on page 296 for a list of defined option names  
/ and option numbers.

## / I.36.2 Operands

/ *option\_number*  
/ is a number from 1 through 254.

/ *option\_value*  
/ is an option value appropriate for the specified option. When an option  
/ statement is present with a DefineOptions level, the value indicates how the  
/ option should be handled when used later in the machine file. In a  
/ DefineOptions level, the option values can be:

### / **ASCII**

/ indicates that the option value will be an ASCII string. A provided value  
/ can be either a single, blank delimited word, or a string (which may contain  
/ blanks), that begins with a single or double quote and ends with the same  
/ quote. The character string is translated from EBCDIC to ASCII before its  
/ use.

/ The data may also be specified as a hexadecimal string which is not  
/ translated. See the HEX value description that follows for information  
/ about how to specify values as hexadecimal strings.

### / **IPADDR**

/ indicates that the allowed option value is an address specified in  
/ dotted-decimal notation (e.g. 9.100.25.100)

### / **IPALIST**

/ indicates that the allowed option value is a list of IP addresses in  
/ dotted-decimal notation, separated by blanks.

### / **HEX**

/ indicates that the allowed option value is a single blank delimited word that  
/ is comprised of pairs of EBCDIC characters (0-1 and A-F) which represent  
/ a hexadecimal byte of data.

### / **UINT(*blen,min,max*)**

/ indicates that the allowed option value is an unsigned whole number.

- *blen* is the length, in octets, of the converted value that will be sent to the client.

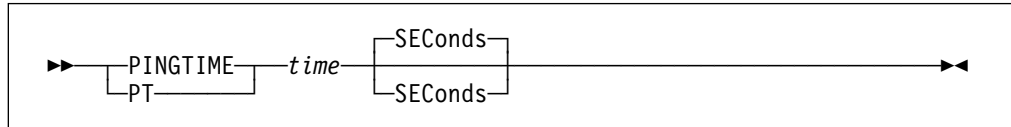


### / I.36.3 Usage Notes

- /
- / 1. The Option statement may be specified at the global, subnet, class, client,  
/ vendor or DefineOptions level only.
- /
- / 2. The following options should not be specified or redefined within the machine  
/ file; they control DHCP processing.
- /
- |           |                             |
|-----------|-----------------------------|
| <b>43</b> | Vendor specific information |
| <b>52</b> | Option overload             |
| <b>53</b> | DHCP message type           |
| <b>54</b> | Server identifier           |
| <b>55</b> | Parameter request list      |
| <b>56</b> | Message                     |
| <b>57</b> | Maximum DHCP message size   |
| <b>60</b> | Class identifier            |
| <b>61</b> | Client identifier           |
| <b>77</b> | Client class                |
- /
- / 3. If an option is not defined, then its value is treated as a character string or, if  
/ the HEX operand is specified, as hexadecimal data.
- /
- / 4. It may be easier to use the DefineOptions statement to define an unrecognized  
/ option than coding the value in ASCII or hexadecimal. For example, if a new  
/ option (for example 190) is used by the client to contain a series of 32 bit  
/ signed time offsets: 18000, 12000, and 3000, you could code it in hex as  
/ follows:
- /
- ```
190 hex 0000465000002ee000000bb8
```
- /
- / or you could define the options and specify the 190 option values as decimal  
/ numbers.
- /
- ```
DefineOptions  
{  
  option 190 lsint(4,*,*)  
}
```
- /
- ```
option 190 18000 12000 3000
```
- /
- / 5. See I.44, "DHCP Options" on page 296 for a description of the recognized  
/ option numbers and the option values which may be specified.

---

## / I.37 PingTime Statement



### / I.37.1 Purpose

/ Use the PingTime statement to specify a time interval that the server will wait for a  
/ response to a ping (ICMP Echo Request). The server pings an IP address before it  
/ is assigned, to make sure that address is not already in use.

### / I.37.2 Operands

/ *time*  
/ is the number of seconds to wait for the response to a ping. The value must  
/ be a whole number (Base 10) between 1 and 30. If this statement is not  
/ specified, the PingTime defaults to 1 second.

#### / **SECOnds**

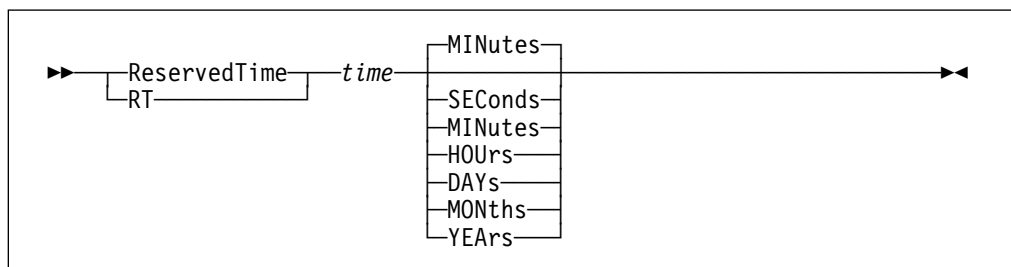
/ indicates that the time value is seconds.

### / I.37.3 Usage Notes

- / 1. This statement should be specified at the global level only and therefore, is  
/ used for all addresses supported by this server.

---

## / I.38 ReservedTime Statement



### / I.38.1 Purpose

/ Use the ReservedTime statement to specify the maximum amount of time the  
/ server holds an offered address in reserve, while waiting for a response from the  
/ client.

## / I.38.2 Operands

/ *time*  
/ is the amount of time to wait for a response from a client that has been offered  
/ an address.

/ The value must be a positive whole number (Base 10) or “-1” (which means  
/ infinity). If this statement is not specified, the ReservedTime defaults to 5  
/ minutes. The minimum time value must be equal to at least 30 seconds.

/ **SEConds**  
/ indicates the time value is in units of seconds.

/ **MINutes**  
/ indicates the time value is in units of minutes.

/ **HOUrs**  
/ indicates the time value is in units of hours.

/ **DAYs**  
/ indicates the time value is in units of days.

/ **MONths**  
/ indicates the time value is in units of months.

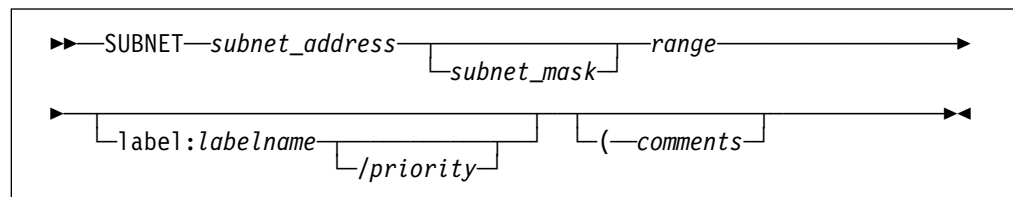
/ **YEArS**  
/ indicates the time value is in units of years.

## / I.38.3 Usage Notes

/ 1. The ReservedTime statement should be specified at the global level only and  
/ therefore, is used for all addresses issued by this server.

---

## / I.39 SUBNET Statement



## / I.39.1 Purpose

/ Use the Subnet statement to specify configuration parameters for an address pool  
/ administered by a server. An address pool is a range of IP addresses to be leased  
/ to clients. The task of configuring subnets also allows you to set lease time and  
/ other options for clients that use the address pool. Lease time and other options  
/ can be inherited from a global level.

## / I.39.2 Operands

/ *subnet\_address*  
/ is the address of this subnet, specified in dotted-decimal notation (for example,  
/ 9.67.48.0).

/ *subnet\_mask*  
/ is the mask for the subnet, in dotted-decimal notation or in integer format. A  
/ subnet mask divides the subnet address into a subnet portion and a host  
/ portion. If no value is entered for the subnet mask, the default is the class  
/ mask appropriate for an A, B, or C class network.

/ A subnet mask can be expressed either in dotted-decimal notation, or as an  
/ integer between 8 and 31. For example, the subnet mask 255.255.240.0 is  
/ expressed in dotted-decimal notation. For subnet 9.67.48.0, this mask implies  
/ an address range from 9.67.48.001 to 9.67.63.254. The equivalent integer  
/ format for this mask is 20; the value 20 is the total number of 1s present when  
/ the mask expressed in binary, as 11111111.11111111.11110000.00000000.

/ Default subnet masks include:

- / • Class A network - 255.0.0.0
- / • Class B network - 255.255.0.0
- / • Class C network - 255.255.255.0

/ *range*  
/ is a range of addresses to be administered to this subnet. Enter the addresses  
/ in dotted-decimal notation, beginning with the lower end of the range, followed  
/ by a hyphen, then the upper end of the range, with no spaces between. For  
/ example, 9.67.48.1-9.67.48.128. If more than one subnet has the same subnet  
/ address and mask, their ranges should not overlap.

/ *labelname*  
/ is a label (a string of 1 to 64 alphanumeric characters) that identifies the subnet  
/ group.

/ *priority*  
/ is the priority that this subnet is used in relation to other subnets which share  
/ the same label.



/ *comments*  
/ anything after the left parenthesis is treated as a comment and ignored by the  
/ server. This allows compatibility with other IBM DHCP servers.

### / **I.39.3 Usage Notes**

- / 1. The Subnet statement should be specified at the global level only.
- / 2. IP addresses specified on the *range* operand may be restricted from general  
/ use in the subnet by CLASS and CLIENT statements that specify IP addresses.  
/ The restricted addresses will be allocated only to clients that belong to the  
/ related class or client.
- / 3. When an address range is specified, do not include the address of the subnet  
/ and the address used for broadcast messages. For example, if the subnet  
/ address is 9.67.96.0 and the subnet mask is 255.255.240.0, do not include  
/ 9.67.96.0 and 9.67.111.255 in the range of addresses.
- / 4. Use the Client statement to exclude an IP address from a range of addresses  
/ that is administered by the DHCP server. For example, exclude an address  
/ that has been permanently assigned to a host. For more information on Client  
/ statements, see I.31, “CLIENT Statement” on page 279.
- / 5. To identify which subnets are grouped together on the same wire, more than  
/ one subnet can have the same identifier. For example, the five subnet  
/ statements that follow are part of two different groups, labelled as WIRE1  
/ (9.67.48.0 and 9.67.49.0) and WIRE3 (9.67.50.0 and 9.67.51.0). Specifying a  
/ label for a subnet group has no affect unless that label is then specified on an  
/ InOrder: or Balance: statement. If not specified on one of these two  
/ statements, a label is ignored during processing, and clients are allocated  
/ addresses for the subnet on which they broadcast their request.  
  
/ The order in which addresses are selected from subnet statements is  
/ determined by the priority associated with each subnet statement. Although  
/ second in the sequence of statements for subnet group WIRE3, the address  
/ pool for subnet 9.67.50.0 (addresses 1-50) is used before the address pool for  
/ subnet 9.67.51.0 — which is defined first in the WIRE3 statement sequence.  
/ The address pool for subnet 9.67.50.0 (addresses 51-100) will be used last for  
/ the WIRE3 subnet group.  
  
/ For subnet group WIRE1 (which is listed on the Balance: statement), the  
/ assigned priority values have less of an effect on the selection of addresses,  
/ because the server will alternate among the members of this group as it  
/ attempts to allocate addresses. In this case, the priority values determine only  
/ the order in which the list of groups is traversed. Here, an attempt is made to  
/ select an address from 9.67.49.0, then 9.67.48.0, then 9.67.49.0, and so on.

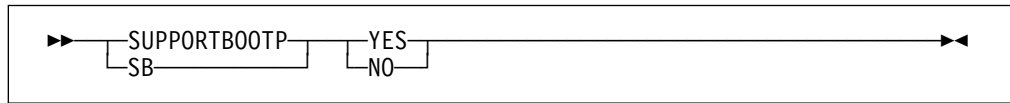
```

/           InOrder: WIRE3
/           Balance: WIRE1
/           subnet 9.67.49.0 255.255.255.0 9.67.49.1-9.67.49.100 label:WIRE1/3
/           subnet 9.67.48.0 255.255.255.0 9.67.48.1-9.67.48.50 label:WIRE1/5
/           subnet 9.67.51.0 255.255.255.0 9.67.51.1-9.67.51.50 label:WIRE3/2
/           subnet 9.67.50.0 255.255.255.0 9.67.50.1-9.67.50.50 label:WIRE3/1
/           subnet 9.67.50.0 255.255.255.0 9.67.50.51-9.67.50.100 label:WIRE3/3

```

---

## / I.40 SupportBootP Statement



### / I.40.1 Purpose

/ Use the SupportBootP statement to specify whether DHCPD responds to requests  
/ from BootP clients.

### / I.40.2 Operands

/ **YES**  
/ indicates that BootP clients should be supported. This is the default if the  
/ statement is not specified.

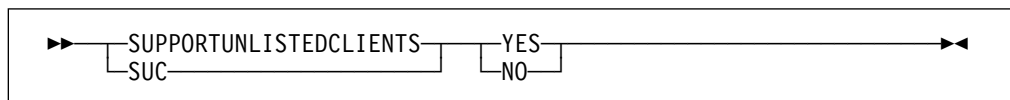
/ **NO**  
/ indicates that BootP clients should not be supported.

### / I.40.3 Usage Notes

- / 1. The SupportBootP statement should be specified at the global level only and  
/ therefore, is used for all BootP requests received by this server.
- / 2. If DHCPD previously supported BootP clients and has been reconfigured to not  
/ support BootP clients, the address binding for any BootP clients that was  
/ established before the reconfiguration will be maintained until the BootP client  
/ sends another request (when it is restarting). At that time, the server will not  
/ respond, and the binding will be removed.

---

## / I.41 SupportUnlistedClients Statement



/ **I.41.1 Purpose**

/ Use the SupportUnlistedClients statement to control whether DHCPD will respond  
/ to clients that are not listed (by client ID) in the machine file.

/ **I.41.2 Operands**

/ **YES**  
/ indicates that clients not listed (by client ID) in the machine file should be  
/ supported.

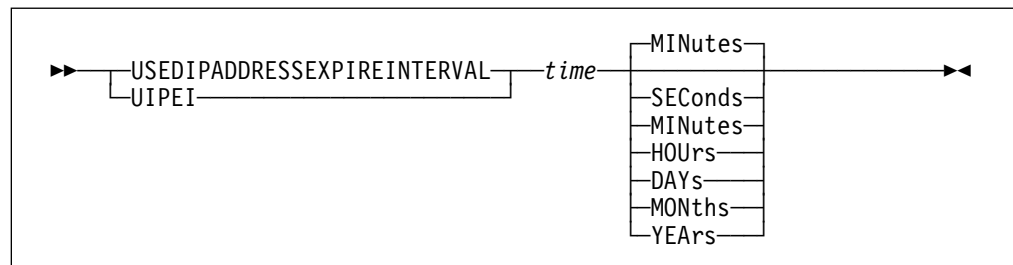
/ **NO**  
/ indicates that only clients listed (by client ID) in the machine file should be  
/ supported.

/ **I.41.3 Usage Notes**

- / 1. The SupportUnlistedClients statement should be specified at the global level  
/ only and therefore, is used for all requests received by this server.
- / 2. This statement can be used to limit access to addresses managed by this  
/ DHCP server. Listing the client IDs for all acceptable clients may be time  
/ consuming.
- / 3. If the SupportUnlistedClients statement is not specified, the default is 'YES' —  
/ unlisted clients are supported.

---

/ **I.42 UsedIPAddressExpireInterval Statement**



/ **I.42.1 Purpose**

/ Use the UsedIPAddressExpireInterval statement to specify the amount of time to  
/ retain an IP address whose lease has expired, before that address is returned to  
/ the available address pool for possible reassignment to another client. This allows  
/ the client that last used an address to again request it — within the specified time  
/ interval — and have that address reassigned to it.

/ **I.42.2 Operands**

/ *time*  
/ is the amount of time to wait before an IP address, whose lease has expired, is  
/ returned to the pool of available addresses.

/ The value must be a positive whole number (Base 10) or "-1" (which means  
/ infinity). If this statement is not specified, the UsedIPAddressExpireInterval  
/ defaults to 30 seconds.

/ **SECONDS**  
/ indicates the time value is in units of seconds.

/ **MINutes**  
/ indicates the time value is in units of minutes.

/ **HOUrs**  
/ indicates the time value is in units of hours.

/ **DAYs**  
/ indicates the time value is in units of days.

/ **MONths**  
/ indicates the time value is in units of months.

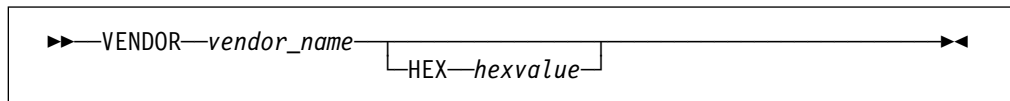
/ **YEArS**  
/ indicates the time value is in units of years.

/ **I.42.3 Usage Notes**

/ 1. The UsedIPAddressExpireInterval statement should be specified at the global  
/ level only and therefore, is used for all addresses supported by this server.

---

/ **I.43 VENDOR Statement**



/ **I.43.1 Purpose**

/ Use the Vendor statement to specify an option 43 value to be returned to clients  
/ that specify an option 60 value that matches the vendor name.

## / I.43.2 Operands

/ *vendor\_name*  
/ is the user-defined label that identifies the vendor. The client transmits this  
/ label using option 60. The vendor name is an ASCII string of up to 255  
/ characters (for example, "IBM"). If the vendor name contains spaces, it must  
/ be surrounded by single (') or double quotes (").

/ **HEX**  
/ indicates that option 43 data follows this keyword.

/ *hexvalue*  
/ is an EBCDIC representation of the hexadecimal value to be returned to the  
/ client in option 43.

## / I.43.3 Usage Notes

- / 1. The Vendor statement should be specified at the global level only.
- / 2. The Vendor statement may be followed by a pair of braces that delimit the  
/ options particular to this vendor. Within these braces, the usual option value  
/ encoding and decoding rules do not apply. An option 43 value will be  
/ constructed from the specified options in the form of an option number,  
/ followed by the length of the data for that option, and the option data.

/ The value for each option must be specified either as an EBCDIC string, or in  
/ hexadecimal form, by specifying the HEX keyword followed by the EBCDIC  
/ representation of the hexadecimal value.

/ For example

```
/ VENDOR IBM  
/ {  
/   OPTION 1 1  
/   OPTION 15 HEX F1  
/ }
```

/ would create an option 43 value of (shown in hex):

/ 0101300F01F1

/ **Note:** The value of 1 for option 1 is '30' which is the ASCII value for the  
/ character '1'.

/ You could also specify the value on the Vendor statement using the hex  
/ keyword, as follows:

```
/ VENDOR IBM hex 0101300F01F1
```

---

## / I.44 DHCP Options

/ The following table lists the option names, their numbers and the locations of  
/ additional information.

/ *Figure 52 (Page 1 of 3). DHCP Option Names and Numbers*

<b>Name</b>	<b>Number</b>	<b>Page</b>
All Subnets are local	27	304
ARP Cache Timeout	35	306
Boot File name	67	311
Boot File Size	13	302
Broadcast Address	28	305
Client Class	77	313
Client Identifier	61	311
Cookie Server	8	301
Default Finder Server	73	313
Default Internet Relay Chat (IRC) Server	74	313
Default IP Time-to-live	23	304
Default World Wide Web (WWW) Server	72	312
DHCP Message Type	53	309
Domain Name	15	302
Domain Name Server	6	300
Ethernet Encapsulation	36	306
Extension Path	18	303
Host Name	12	302
Impress Server	10	301
Interface MTU	26	304
IP Address Lease Time	51	309
IP Forwarding Enable/Disable	19	303
Log Server	7	301
LPR Server	9	301
Mask Supplier	30	305
Maximum Datagram Reassembly Size	22	303

Figure 52 (Page 2 of 3). DHCP Option Names and Numbers

Name	Number	Page
Maximum DHCP Message Size	57	310
Merit Dump File	14	302
Message	56	310
Mobile IP Home Agent	68	312
Name Server	5	300
NetBIOS over TCP/IP Datagram Distribution Server	45	308
NetBIOS over TCP/IP Name Server	44	308
NetBIOS over TCP/IP Node Type	46	308
NetBIOS over TCP/IP Scope	47	308
Network Information Servers	41	307
Network Information Service Domain	40	307
Network Information Service+ Domain	64	311
Network Information Service+ Servers	65	311
Network News Transport Protocol (NNTP) Server	71	312
Network Time Protocol Servers	42	307
Non-Local Source Routing Enable/Disable	20	303
Option Overload	52	309
Parameter Request List	55	310
Path MTU Aging Timeout	24	304
Path MTU Plateau Table	25	304
Perform Mask Discovery	29	305
Perform Router Discovery	31	305
Policy Filter	21	303
Post Office Protocol (POP3) Server	70	312
Rebinding (T2) Time Value	59	310

Figure 52 (Page 3 of 3). DHCP Option Names and Numbers

Name	Number	Page
Renewal (T1) Time Value	58	310
Requested IP Address	50	308
Resource Location Server	11	301
Root Path	17	302
Router Address	3	300
Router Solicitation Address	32	305
Server Identifier	54	309
Simple Mail Transport Protocol (SMTP) Server	69	312
Static Route	33	306
StreetTalk Directory Assistance (STDA) Server	76	313
StreetTalk Server	75	313
Subnet Mask	1	299
Swap Server	16	302
TCP Default TTL	37	306
TCP Keepalive Interval	38	307
TCP Keepalive Garbage	39	307
TFTP Server Name	66	311
Time Offset	2	300
Time Server	4	300
Trailer Encapsulation	34	306
Vendor Class Identifier	60	310
Vendor Specific Information	43	307
X Window System Display Manager	49	308
X Window System Font Server	48	308

Figure 54 on page 299 lists the DHCP option numbers, their description and the type of data which may be entered in the machine file for the option on the Option statement.

The following table lists the complex datatypes that appear in Figure 54 on page 299.



Figure 53. DHCP Option Datatype Formats

Datatype	Format
Character string	A single blank delimited word, or a string (which may contain blanks), that begins with a single or double quote and ends with the same quote. The character string is translated from EBCDIC to ASCII prior to its use.
Hexadecimal string	"HEX" followed by a single blank delimited word containing pairs of EBCDIC characters (0-1, and A-F) which represent a hexadecimal byte of data. The data is assumed to represent ASCII or OCTET values and is not translated from EBCDIC to ASCII as character strings are translated.
IP Address	Dotted-decimal notation (for example, 9.100.25.100)
IP Address List	List of IP addresses separated by blanks
IP Address Pair	Two IP addresses in dotted-decimal notation which are separated by blanks.
IP Address Pair List	List of IP address pairs. All addresses are separated from each other by blanks.
Signed Integer	A single blank delimited word that begins with a sign (+ or -). If a sign is not specified, "+" is the default.
Unsigned Integer	A single blank delimited word that does not begin with a sign and denotes a positive value.
List of Signed Integers	A list of blank delimited numbers that begin with a sign (+ or -). If a sign is not specified, "+" is the default.
List of Unsigned Integers	A list of blank delimited numbers that do not begin with a sign and denote a positive value.

Figure 54 (Page 1 of 15). DHCP Options

Option Number	Option Information
1	<b>Name:</b> SUBNET MASK
	<b>Description:</b> Specifies the client's subnet mask per RFC 950.
	<b>Datatype:</b> IP Address

Figure 54 (Page 2 of 15). DHCP Options

Option Number	Option Information
2	<p><b>Name:</b> Time Offset</p> <p><b>Description:</b> Specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC).</p> <p><b>Datatype:</b> Signed Integer</p> <p><b>Minimum</b> -2,147,483,647</p> <p><b>Maximum</b> 2,147,483,647</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. A positive offset indicates a location east of the zero meridian and a negative offset indicates a location west of the zero meridian.</li> </ol>
3	<p><b>Name:</b> Router Addresses</p> <p><b>Description:</b> Specifies a list of IP address for routers on the client's subnet.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Routers should be listed in order of preference.</li> </ol>
4	<p><b>Name:</b> Time Server</p> <p><b>Description:</b> Specifies a list of RFC 868 time servers available to the client.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
5	<p><b>Name:</b> Name Server</p> <p><b>Description:</b> Specifies a list of IEN 116 name servers available to the client.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
6	<p><b>Name:</b> Domain Name Server</p> <p><b>Description:</b> Specifies a list of Domain Name System (STD 13, RFC 1035) name servers available to the client.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>

Figure 54 (Page 3 of 15). DHCP Options

Option Number	Option Information
7	<b>Name:</b> Log Server
	<b>Description:</b> Specifies a list of MIT-LCS UDP log servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.
8	<b>Name:</b> Cookie Server
	<b>Description:</b> Specifies a list of RFC 865 cookie servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.
9	<b>Name:</b> LPR Server
	<b>Description:</b> Specifies a list of RFC 1179 line printer servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.
10	<b>Name:</b> Impress Server
	<b>Description:</b> Specifies a list of IMAGEN Impress servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.
11	<b>Name:</b> Resource Location Server
	<b>Description:</b> Specifies a list of RFC 886 Resource Location servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.

Figure 54 (Page 4 of 15). DHCP Options

Option Number	Option Information
12	<p><b>Name:</b> Host Name</p> <p><b>Description:</b> Specifies the name of the client.</p> <p><b>Datatype:</b> Character string or hexadecimal string</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. The name may or may not be qualified with the local domain. If the Domain Name option is specified then this option must not include the local domain name as part of the character string.</li> <li>2. See RFC 1035 for character set restrictions.</li> </ol>
13	<p><b>Name:</b> Boot File Size</p> <p><b>Description:</b> Specifies the length (in 512 octet blocks) of the default boot image for the client.</p> <p><b>Datatype:</b> Unsigned Integer</p> <p><b>Maximum</b> 65,535</p>
14	<p><b>Name:</b> Merit Dump File</p> <p><b>Description:</b> Specifies the path name of a file to which the client's core image should be dumped in the event the client crashes.</p> <p><b>Datatype:</b> Character string or hexadecimal string</p>
15	<p><b>Name:</b> Domain Name</p> <p><b>Description:</b> Specifies the domain name that the client should use when resolving host names via the Domain Name System.</p> <p><b>Datatype:</b> Character string or hexadecimal string</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. See the Host Name option for information on specifying these options together.</li> </ol>
16	<p><b>Name:</b> Swap Server</p> <p><b>Description:</b> Specifies the IP address of the client's swap server.</p> <p><b>Datatype:</b> IP Address</p>
17	<p><b>Name:</b> Root Path</p> <p><b>Description:</b> Specifies the path name that contains the client's root disk.</p> <p><b>Datatype:</b> Character string or hexadecimal string</p>

Figure 54 (Page 5 of 15). DHCP Options

Option Number	Option Information
18	<p><b>Name:</b> Extensions Path</p> <p><b>Description:</b> Specifies a file, retrievable using TFTP, that contains information which can be interpreted in the same way as the 64 octet vendor-extension field within the BOOTP response.</p> <p><b>Datatype:</b> Character string or hexadecimal string</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. The file length of the specified file is constrained only by the host file system that contains the file.</li> <li>2. All references to option 18 (this option) within the file are ignored by the client.</li> </ol>
19	<p><b>Name:</b> IP Forwarding Enable/Disable</p> <p><b>Description:</b> Specifies whether the client should configure its IP layer for packet forwarding. A value of 0 means disable IP forwarding, and a value of 1 means enable IP forwarding.</p> <p><b>Datatype:</b> Integer, 0, or 1</p>
20	<p><b>Name:</b> Non-Local Source Routine Enable/Disable</p> <p><b>Description:</b> Specifies whether a client should configure its IP layer to allow forwarding of datagrams with non-local source routes. A value of 0 means that forwarding of such datagrams are not allowed, and a value of 1 means that forwarding is allowed.</p> <p><b>Datatype:</b> Integer, 0, or 1</p>
21	<p><b>Name:</b> Policy Filter</p> <p><b>Description:</b> Specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/ mask pairs with which to filter incoming source routes. Any source routed datagram whose next hop address does not match one of the filters will be discarded by the client.</p> <p><b>Datatype:</b> IP Address Pair List</p>
22	<p><b>Name:</b> Maximum Datagram Reassembly Size</p> <p><b>Description:</b> Specifies the maximum size datagram that the client should be prepared to reassemble.</p> <p><b>Datatype:</b> Unsigned Integer</p> <p><b>Minimum</b> 576</p> <p><b>Maximum</b> 65,535</p>

Figure 54 (Page 6 of 15). DHCP Options

Option Number	Option Information
23	<b>Name:</b> Default IP Time-to-live
	<b>Description:</b> Specifies the default time-to-live that the client should use on outgoing datagrams.
	<b>Datatype:</b> Unsigned integer <b>Minimum</b> 1 <b>Maximum</b> 255
24	<b>Name:</b> Path MTU Aging Timeout
	<b>Description:</b> Specifies the timeout (in seconds) to use, when aging Path MTU values discovered by the mechanism defined in RFC 1191.
	<b>Datatype:</b> Unsigned integer <b>Maximum</b> 4,294,967,295
25	<b>Name:</b> Path MTU Plateau Table
	<b>Description:</b> Specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest.
	<b>Datatype:</b> List of unsigned integers <b>Minimum</b> 68 <b>Maximum</b> 65,535
26	<b>Name:</b> Interface MTU
	<b>Description:</b> Specifies the MTU to use on this interface.
	<b>Datatype:</b> Unsigned integer <b>Minimum</b> 68 <b>Maximum</b> 65,535
27	<b>Name:</b> All Subnets are Local
	<b>Description:</b> Specifies whether the client may assume that all subnets of the IP network use the same MTU as the subnet of that network to which the client is directly connected.
	<p>A value of 1 means that all subnets share the same MTU. A value of 0 means that the client should assume that some subnets of the directly connected network may have smaller MTUs.</p>
<b>Datatype:</b> Integer, 0, or 1	

Figure 54 (Page 7 of 15). DHCP Options

Option Number	Option Information
28	<p><b>Name:</b> Broadcast Address</p> <p><b>Description:</b> Specifies the broadcast address in use on the client's subnet.</p> <p><b>Datatype:</b> IP Address</p> <p><b>Notes:</b> Legal values for broadcast addresses are specified in STD 3, RFC 1122.</p>
29	<p><b>Name:</b> Perform Mask Discovery</p> <p><b>Description:</b> Specifies whether the client should perform subnet mask discovery using ICMP. A value of 0 indicates that the client should not perform mask discovery. A value of 1 means that the client should perform mask discovery.</p> <p><b>Datatype:</b> Integer, 0, or 1</p>
30	<p><b>Name:</b> Mask Supplier</p> <p><b>Description:</b> Specifies whether the client should respond to subnet mask requests using ICMP. A value of 0 indicates that the client should not respond. A value of 1 means that the client should respond.</p> <p><b>Datatype:</b> Integer, 0, or 1</p>
31	<p><b>Name:</b> Perform Router Discovery</p> <p><b>Description:</b> Specifies whether the client should solicit routers using the Router Discover mechanism defined in RFC 1256. A value of 0 indicates that the client should not perform router discovery. A value of 1 means that the client should perform router discovery.</p> <p><b>Datatype:</b> Integer, 0, or 1</p>
32	<p><b>Name:</b> Router Solicitation Address</p> <p><b>Description:</b> Specifies the IP address to which the client should transmit router solicitation requests.</p> <p><b>Datatype:</b> IP Address</p>

Figure 54 (Page 8 of 15). DHCP Options

Option Number	Option Information
33	<p><b>Name:</b> Static Route</p> <p><b>Description:</b> Specifies a list of static routes that the client should install in its routing cache. The routes consist of pairs of IP addresses. The first address is the destination address, and the second address is the router for the destination.</p> <p><b>Datatype:</b> IP Address Pair List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. If multiple routes to the same destination are specified, they should be listed in descending order of priority.</li> <li>2. The default route (0.0.0.0) is an illegal destination for a static route.</li> </ol>
34	<p><b>Name:</b> Trailer Encapsulation</p> <p><b>Description:</b> Specifies whether the client should negotiate the use of trailers (RFC 893) when using the ARP protocol. A value of 0 indicates that the client should not attempt to use trailers. A value of 1 means that the client should attempt to use trailers.</p> <p><b>Datatype:</b> Integer, 0, or 1</p>
35	<p><b>Name:</b> ARP Cache Timeout</p> <p><b>Description:</b> Specifies the timeout in seconds for ARP cache entries.</p> <p><b>Datatype:</b> Unsigned Integer</p> <p><b>Maximum</b> 4,294,967,295</p>
36	<p><b>Name:</b> Ethernet Encapsulation</p> <p><b>Description:</b> Specifies whether the client should use Ethernet Version 2 (RFC 894) or IEEE 802.3 (RFC 1042) encapsulation if the interface is an Ethernet. A value of 0 indicates that the client should use RFC 894 encapsulation. A value of 1 indicates that the client should use RFC 1042 encapsulation.</p> <p><b>Datatype:</b> Integer, 0, or 1</p>
37	<p><b>Name:</b> TCP Default TTL</p> <p><b>Description:</b> Specifies the default TTL that the client should use when sending TCP segments.</p> <p><b>Datatype:</b> Unsigned integer</p> <p><b>Maximum</b> 255</p>



Figure 54 (Page 9 of 15). DHCP Options

Option Number	Option Information
38	<p><b>Name:</b> TCP Keepalive Interval</p> <p><b>Description:</b> Specifies the interval (in seconds) that a TCP client should wait before sending a keepalive message on a TCP connection. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested to do so by an application.</p> <p><b>Datatype:</b> Unsigned Integer</p> <p><b>Maximum</b> 4,294,967,295</p>
39	<p><b>Name:</b> TCP Keepalive Garbage</p> <p><b>Description:</b> Specifies whether the client should send TCP keepalive messages with an octet of garbage for compatibility with older implementations. A value of 0 indicates that a garbage octet should not be sent. A value of 1 indicates that a garbage octet should be sent.</p> <p><b>Datatype:</b> Integer, 0, or 1</p>
40	<p><b>Name:</b> Network Information Service Domain</p> <p><b>Description:</b> Specifies the name of the client's NIS domain.</p> <p><b>Datatype:</b> Character string or hexadecimal string</p>
41	<p><b>Name:</b> Network Information Servers</p> <p><b>Description:</b> Specifies a list of IP addresses indicating NIS servers available to the client.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
42	<p><b>Name:</b> Network Time Protocol Servers</p> <p><b>Description:</b> Specifies a list of IP addresses indicating NTP servers available to the client.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
43	<p><b>Name:</b> Vendor Specific Information</p> <p><b>Description:</b> Specifies hexadecimal data to return to the client. This information is vendor specific.</p> <p><b>Notes:</b></p> <p><b>RESTRICTED.</b> This option may not be specified. It is generated by the Vendor statement only.</p>

Figure 54 (Page 10 of 15). DHCP Options

Option Number	Option Information
44	<p><b>Name:</b> NetBIOS over TCP/IP Name Server</p> <p><b>Description:</b> Specifies a list of RFC 1001/1002 NBNS name servers.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
45	<p><b>Name:</b> NetBIOS over TCP/IP Datagram Distribution Server</p> <p><b>Description:</b> Specifies a list of RFC 1001/1002 NBDD servers.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <p>Servers should be listed in order of preference.</p>
46	<p><b>Name:</b> NetBIOS over TCP/IP Node Type</p> <p><b>Description:</b> Specifies the client type.</p> <p><b>Datatype:</b> Integer, 1, 2, 4 or 8</p>
47	<p><b>Name:</b> NetBIOS over TCP/IP Scope</p> <p><b>Description:</b> Specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002.</p> <p><b>Datatype:</b> Character string or hexadecimal string</p>
48	<p><b>Name:</b> X Window System Font Server</p> <p><b>Description:</b> Specifies a list of X Window System Font Servers available to the client.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
49	<p><b>Name:</b> X Window System Display Manager</p> <p><b>Description:</b> Specifies a list of IP address of systems that are running the X Window System Display Manager and are available to the client.</p> <p><b>Datatype:</b> IP Address List</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
50	<p><b>Name:</b> Network Information Service+ Domain</p> <p><b>Description:</b> Specifies the name of the client's NIS+ domain.</p> <p><b>Datatype:</b> Character string or hexadecimal string</p>

Figure 54 (Page 11 of 15). DHCP Options

Option Number	Option Information
51	<p><b>Name:</b> IP Address Lease Time</p> <p><b>Description:</b> Specifies maximum default lease time in seconds. This option is used in a server reply (DHCPOFFER) to specify the lease time the server is willing to offer.</p> <p><b>Datatype:</b> Unsigned integer</p> <p><b>Maximum</b> 4,294,967,295</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. This option overrides the lease time specified by the DHCPD LEASETIMEDEFAULT statement.</li> </ol>
52	<p><b>Name:</b> Option Overload</p> <p><b>Description:</b> Specifies which fields are being overloaded to carry DHCP options. A DHCP server inserts this option if the returned parameters will exceed the usual space allowed for options.</p> <p><b>1</b> means that the 'file' field is used to hold options.</p> <p><b>2</b> means that the 'sname' field is used to hold options.</p> <p><b>3</b> means that both the 'file' and 'sname' field are used to hold options.</p> <p><b>Notes:</b></p> <p><b>RESTRICTED.</b> This option may not be specified. It is generated by the server to control processing.</p>
53	<p><b>Name:</b> DHCP Message Type</p> <p><b>Description:</b> Specifies the type of DHCP message being sent.</p> <p><b>Notes:</b></p> <p><b>RESTRICTED.</b> This option may not be specified. It is generated by the server and the client to control processing.</p>
54	<p><b>Name:</b> Server Identifier</p> <p><b>Description:</b> Specifies the IP address of the DHCP server. DHCP Servers use this option to distinguish their DHCPOFFER responses from other servers' responses. DHCP clients use the contents of this option as the destination address for any DHCP message unicast to the server. They also use this option to indicate which server's offer is being accepted.</p> <p><b>Notes:</b></p> <p><b>RESTRICTED.</b> This option may not be specified. It is generated by the server and the client to control processing.</p>

Figure 54 (Page 12 of 15). DHCP Options

Option Number	Option Information
55	<p><b>Name:</b> Parameter Request List</p> <p><b>Description:</b> Specifies a list of options requested by the client and the order which the client would like them to appear.</p> <p><b>Notes:</b>  <b>RESTRICTED.</b> This option may not be specified. It is generated by the server and the client to control processing.</p>
56	<p><b>Name:</b> Message</p> <p><b>Description:</b> Specifies an error message to be passed between the client and the server.</p> <p><b>Notes:</b>  <b>RESTRICTED.</b> This option may not be specified. It is generated by the server and the client to control processing.</p>
57	<p><b>Name:</b> Maximum DHCP Message Size</p> <p><b>Description:</b> Specifies the maximum length DHCP message that the client is willing to receive.</p> <p><b>Notes:</b>  <b>RESTRICTED.</b> This option may not be specified. It is generated by the client to control processing.</p>
58	<p><b>Name:</b> Renewal (T1) Time Value</p> <p><b>Description:</b> Specifies the time interval (in seconds) from address assignment until the client transitions to the renewing state.</p> <p><b>Datatype:</b> Unsigned Integer  <b>Maximum</b> 4,294,967,295</p>
59	<p><b>Name:</b> Rebinding (T2) Time Value</p> <p><b>Description:</b> Specifies the time interval (in seconds) from address assignment until the client transitions to the rebinding state.</p> <p><b>Datatype:</b> Unsigned Integer  <b>Maximum</b> 4,294,967,295</p>
60	<p><b>Name:</b> Vendor Class</p> <p><b>Description:</b> Specifies a string which identifies the class of vendor. This information is compared to the value specified on the Vendor machine statement as the vendor name.</p> <p><b>Notes:</b>  <b>RESTRICTED.</b> This option may not be specified. It is generated by the client to control processing.</p>

Figure 54 (Page 13 of 15). DHCP Options

Option Number	Option Information
61	<b>Name:</b> Client Identifier
	<b>Description:</b> Specifies a unique identifier for the client.
	<b>Notes:</b> <b>RESTRICTED.</b> This option may not be specified. It is generated by the client to control processing.
64	<b>Name:</b> Network Information Service+ Domain
	<b>Description:</b> Specifies the name of the client's NIS+ domain.
	<b>Datatype:</b> Character string or hexadecimal string
65	<b>Name:</b> Network Information Service+ Servers
	<b>Description:</b> Specifies a list of IP addresses indicating NIS+ servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be specified in order of preference.
66	<b>Name:</b> TFTP Server Name
	<b>Description:</b> Specifies the name of the TFTP server which the client should use.
	<b>Datatype:</b> Character string or hexadecimal string
	<b>Notes:</b> 1. This value is passed to the client in the 'sname' field or as an option, if option overloading occurs. See option 52, for information on overloading.
67	<b>Name:</b> Bootfile name
	<b>Description:</b> Specifies the name of the bootfile for the client to request.
	<b>Datatype:</b> Character string or hexadecimal string
	<b>Notes:</b> 1. This value is passed to the client in the 'file' field or as an option, if option overloading occurs. See option 52, for information on overloading.

Figure 54 (Page 14 of 15). DHCP Options

Option Number	Option Information
68	<b>Name:</b> Mobile IP Home Agent
	<b>Description:</b> Specifies a list of IP addresses indicating mobile IP home agents available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> <ol style="list-style-type: none"> <li>1. Agents should be listed in order of preference.</li> <li>2. A 0 may be specified instead of an IP address list to indicate that no home agents are available.</li> </ol>
69	<b>Name:</b> Simple Mail Transport Protocol (SMTP) Server
	<b>Description:</b> Specifies a list of IP addresses indicating SMTP servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
70	<b>Name:</b> Post Office Protocol (POP3) Server
	<b>Description:</b> Specifies a list of IP addresses indicating POP3 servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
71	<b>Name:</b> Network News Transport Protocol (NNTP) Server
	<b>Description:</b> Specifies a list of IP addresses indicating NNTP servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>
72	<b>Name:</b> Default World Wide Web (WWW) Server
	<b>Description:</b> Specifies a list of IP addresses indicating WWW servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> <ol style="list-style-type: none"> <li>1. Servers should be listed in order of preference.</li> </ol>

Figure 54 (Page 15 of 15). DHCP Options

Option Number	Option Information
73	<b>Name:</b> Default Finger Server
	<b>Description:</b> Specifies a list of IP addresses indicating Finger servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.
74	<b>Name:</b> Default Internet Relay Chat (IRC) Server
	<b>Description:</b> Specifies a list of IP addresses indicating IRC servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.
75	<b>Name:</b> StreetTalk Server
	<b>Description:</b> Specifies a list of IP addresses indicating StreetTalk servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.
76	<b>Name:</b> StreetTalk Directory Assistance (STDA) Server
	<b>Description:</b> Specifies a list of IP addresses indicating STDA servers available to the client.
	<b>Datatype:</b> IP Address List
	<b>Notes:</b> 1. Servers should be listed in order of preference.
77	<b>Name:</b> Client Class
	<b>Description:</b> Specifies a character string which identifies the client's class. Option 77 is compared to the <i>class_name</i> value specified on the machine file Class statement.
	<b>Notes:</b> <b>RESTRICTED.</b> This option may not be specified. It is generated by the client to control processing.

---

## Appendix J. Configuring the TFTP Virtual Machine

**Note!**

This appendix consolidates information previously found in the TFTP README file provided with APAR PQ01770, and the TCP/IP V2 R4 for VM Corrective Service Memo.

The TFTP daemon (the TFTP server) transfers files between the Byte File System (BFS) and TFTP clients. TFTP supports access to files maintained in a Byte File System directory structure that is mounted during initialization.

To configure the TFTP virtual machine, you must perform the following steps:

**TFTP Configuration Steps**

1. Update the TCPIP server configuration file.
2. Update the TFTP profile exit.
3. Review and address additional configuration considerations.
4. Create the TFTP PERMLIST data file.
5. Create the TFTP USERLIST data file.

This appendix describes the files used to configure TFTP server, as well as the TFTP command and subcommands which control its operation.

---

### J.1 Update the TCPIP Server Configuration File

You should include TFTP in the AUTOLOG statement of the TCPIP server configuration file (PROFILE TCPIP) so that the TFTP virtual machine will be automatically started when TCPIP is invoked. Verify that the following statements are added to the PROFILE TCPIP file:

```
AUTOLOG
  TFTP 0
```

By default, the TFTP server listens on port 69; however, this default can be changed through additional TFTP customization. Verify that the following statement is added to your TCPIP server configuration file as well:

```
PORT
  69 UDP TFTP ; TFTP Server
```

If you customize the port at which the TFTP server listens, ensure you specify your value in place of 69 when you update the PORT statement.



You should also verify that the Maximum Transmission Unit (MTU) size — specified on the GATEWAY statements for the subnetworks on which your IBM Network Stations will operate — is as large as can be reasonably be specified for your environment. Ideally, the MTU size should be greater than or equal to the block size used by TFTP to load files to the IBM Network Station (by default, 8192 bytes), so that high performance is attained when files are transferred using TFTP.

However, you should use care when you specify or change the MTU size for any network or subnetwork. You should review Chapter 25, “Tuning Performance,” of the *TCP/IP V2 R4 for VM: Planning and Customization* prior to making changes of this nature.

---

## J.2 Update the TFTP Profile Exit

The TFTP PROFILE EXEC invokes the TFTPDXIT EXEC customization exit (if it exists) and stacks the TCPRUN EXEC for execution. TCPRUN then starts the TFTP server using the TCPRUN global variables OWNER, COMMAND, and PARMS. You can use the TFTPDXIT customization exit to alter these variables to change the console log owner, the server module name, or the parameters passed to the server module. See Chapter 5, “Server Profile Exits” of *TCP/IP V2 R4 for VM: Planning and Customization* for more information about using the customization exits.

**Note:** You must modify the TFTPDXIT EXEC if:

- You are using a server module that is not located on the TCPIP server minidisk (TCPMAINT 591).
- You change parameters passed to the TFTP command.
- You change the user ID of the virtual machine that receives the TFTP console output.

**Note:** If no parameters are specified in the TFTPDXIT customization exit, TFTP is initialized with the following operands:

```
69 ( SVMNAME TCPIP XFERMODE OCTET
```

---

## J.3 Additional TFTP Configuration Considerations

### J.3.1 Restricting TFTP Access to Files

If you intend to restrict TFTP access to certain files maintained in the Byte File System (BFS), or make use of the TFTP **CREATION** function, you will need to ensure the TFTP server is configured as a POSIX “superuser.” To allow this capability, the following statement must be included in the TFTP user ID directory entry:

POSIXINFO UID 0 GID 0

See the *OpenEdition for VM/ESA User's Guide* and *VM/ESA Planning and Administration* for more information about configuring the TFTP server in this manner.

### J.3.2 Changing the Default TFTP BFS Directory

Because the TFTP server provides BFS-specific file support, a BFS directory must be mounted by this server during initialization so that the necessary OpenEdition Shell and Utilities commands are available. By default, this directory is:

```
./VMBFS:VMSYS:ROOT/
```

If the OpenEdition Shell and Utilities Feature for VM/ESA has been installed in a non-default file pool (that is, different from VMSYS), you will need to change the default mount directory to reflect the file pool used for your root file space and root directory. This can be done through use of the BFSDIR global variable, defined in the TFTPDXIT EXEC customization exit.

### J.3.3 Controlling TFTP Data Translation

The IBM Network Station support provided by APAR PQ02301 requires the TFTP server to not perform any EBCDIC to NETASCII translation of the data it transfers. Thus, the **XFERMODE OCTET** operand must be specified when TFTP is initialized; this is the default. You can change this default through use of the PARMS global variable, defined in the TFTPDXIT EXEC customization exit.

### J.3.4 Collecting TFTP Monitor Data

The TFTP server creates monitor data during its operation. To allow this data to be collected by CP, the APPLMON option must be included in the TFTP directory entry. See Appendix O, "TFTP APPLDATA Monitor Records" on page 360 for more information about the monitor records created by TFTP.

---

## J.4 Create the TFTP PERMLIST Data File

The TFTP PERMLIST file identifies files which the TFTP server should maintain in cache, after an initial access to one of these files by a client. The TFTP PERMLIST file format is:

- One line per pathname.
- Blank lines and lines where the first word is a "\*" are ignored by TFTP.

The TFTP PERMLIST file should be created and maintained on the TCPMAINT 198 minidisk.

A sample TFTP PERMLIST file follows:

```

* -----
* Sample TFTPd PERMLIST file
* -----
* This file identifies files which are to be maintained in cache by the
* TFTPd server after an initial access by a client.
*
  /Xncd
  /mods/login.nws
  /mods/show.nws

```

**Notes:**

1. Maintaining files in memory improves TFTPd performance at the expense of increased virtual storage utilization. You may need to increase virtual storage for the TFTPd server to account for the number and size of the files listed in the TFTPd PERMLIST file.
2. Absolute path names should be used to identify files in the TFTPd PERMLIST file.
3. For more information about how to manage which files TFTPd maintains in cache, see J.16, "LOADPERM Subcommand" on page 328 and J.12, "DROPFILE Subcommand" on page 326

---

## J.5 Create the TFTPd USERLIST Data File

The TFTPd USERLIST file associates client IP addresses with a VM system user ID. This mapping allows an administrator to control user access to various files via the POSIX user ID (UID) and group ID (GID) values that correspond to the VM user IDs identified within this file. The TFTPd USERLIST file format is:

- One line per IP address and user ID pair. This line consists of two blank delimited words, where the:
  - first word is the client IP address, in dotted decimal notation.
  - second word is the VM user ID that should be associated with this IP address. A user ID may be associated with more than one client IP address.
- Blank lines and lines where the first word is a "\*" are ignored by TFTPd.

The TFTPd USERLIST file should be created and maintained on the TCPMAINT 198 minidisk.

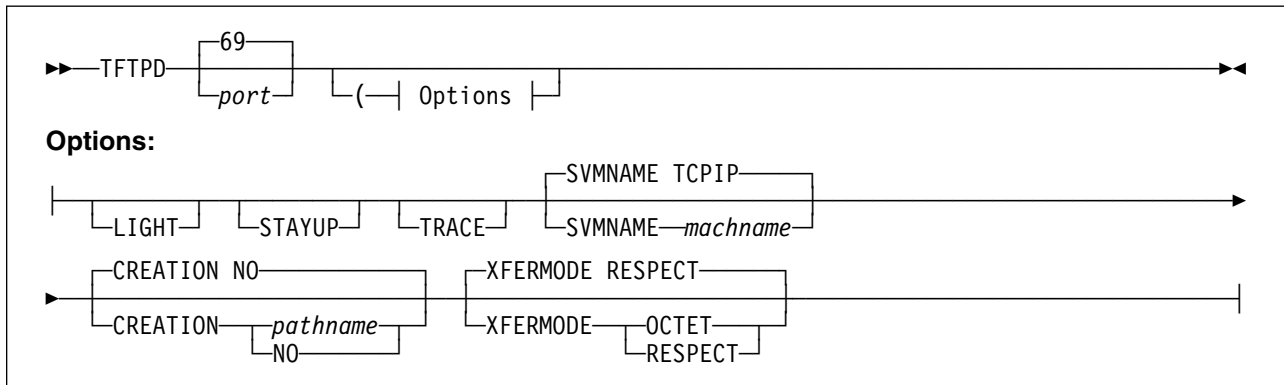
A sample TFTPd USERLIST file follows:

```

* -----
* Sample TFTP USERLIST file
* -----
* This file maps client IP addresses to VM system user IDs.
* IP addresses within this file must be specified using dotted decimal
* notation (for example, 123.45.67.89).
*
9.130.21.12 GEDDYLEE
9.130.21.33 ROCKTMAN
9.130.21.61 DYLANBOB
9.130.21.19 STELYDAN

```

## J.6 TFTP Command



### J.6.1 Purpose

The TFTP daemon (the TFTP server) transfers files between the Byte File System (BFS) and TFTP clients. TFTP supports access to files maintained in a Byte File System directory structure that is mounted during initialization. Files may be specified as absolute or relative path names; fully-qualified path names are rejected.

### J.6.2 Operands

*port*

is the port on which the TFTP daemon should listen for requests.

#### **CREATION NO**

indicates that clients may not create files in the Byte File System.

#### **CREATION *pathname***

indicates that clients may create files in the Byte File System on the specified directory or subdirectories built off of that directory; *pathname* may be an absolute path name or a relative path name of an existing directory in the Byte File System. See the "Usage Notes" section in J.11, "CREATION subcommand" on page 324 for additional information about this function.

|  
|  
| **LIGHT**

| indicates that the TFTP daemon should operate in light-traffic mode. See J.15,  
| “LIGHT Subcommand” on page 327 for further information about this mode.  
|

| **STAYUP**

| indicates that the TFTP daemon should continue to operate across VM TCP/IP  
| failures.  
|

| **TRACE**

| indicates that the TFTP daemon should display debug information as requests  
| are processed.  
|

| **SVMNAME** *machname*

| controls the name of the TCP/IP server machine with which the TFTP daemon  
| should interact.  
|

| **XFERMODE RESPECT**

| indicates that TFTPd should respect the transfer mode requested by the client.  
| All data in the Byte File System is assumed to be EBCDIC data. When the  
| client requests that the file be transferred as NETASCII, the data is converted  
| between EBCDIC and NETASCII.  
|

| **XFERMODE OCTET**

| indicates that TFTPd should ignore the transfer mode requested by the client.  
| No translation will be performed on the transferred data. This allows data to be  
| stored in the Byte File System in ASCII format.  
|

| **J.6.3 Usage Notes**

- | 1. The TFTPd server used to provide TFTP support does not need reside on the  
| VM system that participates in the subnet in which this activity occurs.  
| However, for performance reasons, IBM recommends this server to be present  
| there, as this eliminates the need for router involvement in order to forward  
| information between the client and the VM system where the TFTPd server is  
| located.  
|
- | 2. TFTPd reads files into memory (cache) before they are sent to a client.  
| Similarly, it receives a complete file before that file is written to the Byte File  
| System. In addition, the permanent file list (TFTPd PERMLIST) identifies files  
| that should be kept in memory. Due to the fact that files are maintained in  
| memory, you may need to increase the virtual machine size to accommodate  
| the number and size of files that your machine handles. Errors due to  
| insufficient storage in the TFTPd virtual machine will cause read or write  
| operations to fail.  
|
- | 3. The **STAYUP** option is needed only when the TCP/IP machine does not  
| contain an entry for the virtual machine running TFTPd.  
|

## TFTPD Subcommands

### J.6.4 Messages

Error messages may include:

- Unrecognized port number.
- Option *option* requires a value.
- Unrecognized option *option* - continuing
- Unrecognized operand *operand* -- continuing
- Pathname specified for CREATION is not valid
- Unrecognized value for option XFERMODE

Return codes include:

- 108** A port number that is not valid was specified on the TFTP command.
- 109** A multitasking setup function failed (Queue Create).
- 110** A multitasking setup function failed (Queue Identify Service).
- 111** A multitasking setup function failed (Thread Create).
- 112** A multitasking setup function failed (Semaphore Create).
- 114** An unrecognized service machine name was specified on the TFTP command.
- 118** Unable to read the TFTP USERLIST file.
- 119** Unable to open the Permanent file list for reading.
- 120** Unable to read the TFTP PERMLIST file.
- 122** Error detected with the specified XFERMODE option.
- 123** Error detected with the specified CREATION option.

---

### J.7 TFTP Subcommands

You must be logged on to the TFTP server to use the TFTP subcommands. The TFTP subcommands are listed in Figure 55. This table provides the shortest abbreviation, a description, and a page reference for more information for each TFTP subcommand.

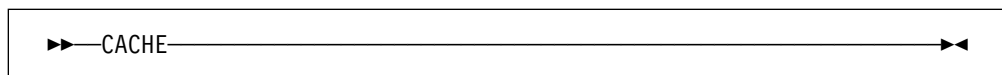
Figure 55 (Page 1 of 2). TFTP Subcommands

Subcommand	Minimum Abbreviation	Description	Page
CACHE	CACHE	Displays a list of cached files.	321
CLIENTS	CLIENTS	Displays a list of clients being served by TFTP.	323
CMS	CMS	Passes a command to CMS for execution.	324
CREATION	CREATION	Specifies a directory on which clients may create files.	324
DROPPFILE	DROPPFILE	Removes a specific file from cache.	326
EXIT	EXIT	Stops the TFTP server and its processing. EXIT is equivalent to QUIT and STOP.	327

Figure 55 (Page 2 of 2). TFTP Subcommands

Subcommand	Minimum Abbreviation	Description	Page
HELP	HELP	Displays a summary of TFTP subcommands.	327
LIGHT	LIGHT	Toggles LIGHT processing mode of the TFTP server.	327
LOADPERM	LOADPERM	Loads the TFTP PERMLIST file, which lists files to be maintained in cache.	328
LOADUSER	LOADUSER	Loads the user mapping file, TFTP USERLIST.	328
QUIT	QUIT	Stops the TFTP server and its processing. QUIT is equivalent to EXIT and STOP.	329
STAYUP	STAYUP	Toggles the STAYUP mode of the TFTP server.	329
STOP	STOP	Stops the TFTP server and its processing. STOP is equivalent to EXIT and QUIT.	330
TRACE	TRACE	Toggles the TRACE mode of the TFTP server.	330
XFERMODE	XFERMODE	Determines whether NETASCII file translation should be performed.	331

## J.8 CACHE Subcommand



### J.8.1 Purpose

Use the CACHE subcommand to display the list of cached files.

### J.8.2 Usage Notes

1. Once a file is marked to be dropped from cache via the **DROFFILE** subcommand, the next new request by a client to read that file will cause a new version of the file to be obtained from the Byte File System.
2. If a client's request for a file "can be satisfied" from memory, such is done. Otherwise, a copy of the file is loaded from DASD into memory and that copy is used.

The phrase "can be satisfied" is true if:

- the requested file is in memory, and
- it is in the correct format (either NETASCII or OCTET), and
- it is not marked to be dropped by the **DROFFILE** subcommand.

It is possible that more than one client may use the same memory image of a given file, even if that file is not in the permanent file list (the TFTP PERMLIST file). This could happen if the timing is such that a file already present in memory to satisfy one client's request can also satisfy the new client's request.

3. A file is dropped from cache when no clients are using it, unless it is marked as "permanent," by appearing in the permanent file list.

4. To keep a file in memory even though no clients are using it, the file must be listed in the TFTP PERMLIST file.

The permanent file list is loaded when the TFTP daemon is initialized, or when the **LOADPERM** subcommand is issued.

5. A file is read from DASD only when it is requested by a client. There is no provision to "pre-load" a file into cache before it is requested by a client.

### J.8.3 Responses

1. When at least one file is cached, a table of cached files is displayed. This table has the following format:

```
PD InProg M Filesize Pathname
pd iiii m filesize pathname
```

where:

*pd* is a two digit field with a 1 or 0 possible for each position, indicating "yes" or "no," respectively.

- The first digit, *p*, indicates whether the file is being maintained permanently in storage (marked as "permanent").
- The second digit *d*, indicates whether the file will be dropped from cache once its current use count goes to zero.<sup>1</sup>

If *d* is 1, the next new request by a client to read this file will cause the TFTP daemon to obtain a new copy from the Byte File System. If *d* is 0, this file will be dropped from cache as soon as its use count goes to zero.

*iiii* is the number of clients to which the file is being sent.

*m* is the transfer mode, either "n" for NETASCII, or "o" for OCTET.

*filesize*

is the size of the file, in bytes.

---

<sup>1</sup> This information is useful only if the value of *p* is 1, indicating that the file is in the permanent file list.



| *pathname*

| is the name of the file.

| 2. When no files are maintained in memory, the following message is displayed:

| No files cached.

---

## J.9 CLIENTS Subcommand

▶▶—CLIENTS—◀◀

### J.9.1 Purpose

Use the CLIENTS subcommand to display the list of clients that are currently being served by TFTP.

### J.9.2 Responses

1. When at least one client is being served, a table of such clients is displayed.

This table has the following format:

Client	Userid	Bytes	BlkSz	DM	Pathname
<i>ipaddr</i>	<i>userid</i>	<i>bytes</i>	<i>blksz</i>	<i>dm</i>	<i>pathname</i>

*ipaddr*

is the IP address of the client being served; this address is shown in dotted decimal notation (for example, xxx.xxx.xxx.xxx).

*userid*

is the user ID associated with this client. If the client is not associated with any user ID, this field is blank.

*bytes*

is the number of bytes transferred thus far between this client and the TFTP daemon.

*blksz*

is the block size being used for data transfer to this client.

*dm*

is a two character field indicating the direction of file transfer and the transfer mode.

The first character, *d*, is the direction of transfer. *d* will have the value “w” if the transfer is from the IBM Network Station to the Byte File System (a client write request); “r” is used to indicate a transfer from the Byte File System to the client (a client read request).

The second character, *m*, is the transfer mode, either “n” for NETASCII, or “o” for OCTET.

| *pathname*

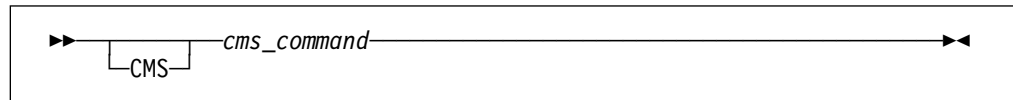
| is the name of the file being transferred.

| 2. When no clients are being served, the following message is displayed:

| No clients found.

---

## J.10 CMS Subcommand



### J.10.1 Purpose

Use the CMS subcommand to issue a command to CMS.

### J.10.2 Operands

*cms\_command*

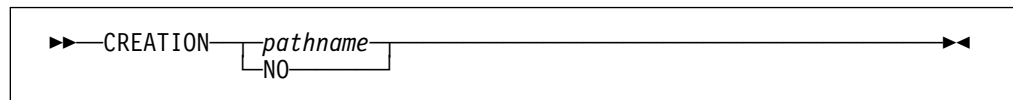
is the CMS command to be issued.

### J.10.3 Usage Notes

1. Do not issue any CMS command that would take considerable time to execute (for example, XEDIT). While the CMS command executes, the server does not respond to requests.
2. The **CMS** keyword is usually not required because the daemon will pass any command string that is not recognized as a TFTP subcommand to CMS. The **CMS** keyword is used to identify commands which would normally be interpreted as a subcommand, for example, **TRACE**.

---

## J.11 CREATION subcommand



### J.11.1 Purpose

Use the CREATION subcommand to specify a directory on which clients may create files.

## J.11.2 Operands

*pathname*

is the name of the directory on which files may be created. In addition, files may be created on any subdirectory of the specified directory; *pathname* may be an absolute path name or a relative path name of an existing directory in the byte file system.

**NO**

disables the ability for clients to create files.

## J.11.3 Usage Notes

1. Fully-qualified path names are not permitted for *pathname*.

2. If *pathname* is specified as a relative path name (does not begin with a slash—"/"), the pathname of the creation directory is constructed each time a write request is processed for a file that does not exist.

The OpenEdition for VM/ESA **OPENVM SET DIRECTORY** command allows you to change the current working directory. The current working directory is used when resolving a relative address into an absolute address.

3. The TFTP virtual machine must be a POSIX "superuser" in order to create files for clients.

4. In addition to the request to create a file on the creation directory or a subdirectory, a client must have write permission for the target directory for the file on a client's behalf.

5. Files created by TFTP are assigned the POSIX user ID (UID) and group ID (GID) of the VM user ID that corresponds to the client, as defined in the TFTP USERLIST file. If the client is not associated with a VM user ID, the created file is assigned a POSIX UID of -1 and a GID of -1.

6. The permission assigned to a created file depends upon the POSIX UID and GID values associated with the client's corresponding VM user ID:

- Owner permission is set according to the corresponding user ID UID value. If the UID is -1, no Owner permission bits are turned on; otherwise, Owner permission is set to read and write (rw-).
- Group permission is set according to the corresponding user ID GID value. If the GID is -1, no Group permission bits are turned on; otherwise, Group permission is set to read and write (rw-).
- Other permission is set according to the corresponding user ID UID value. If the UID is -1, Other permission is set to read and write; otherwise, Other permission is neither read, write, nor executable (---).

The OpenEdition for VM/ESA **OPENVM SET MASK** command may be used to further restrict the permissions that are set when a file is created; this command

specifies permissions which are to be “screened out.” For example, the **OPENVM SET MASK** command could be added to the TFTP profile exit (TFTPD EXIT EXEC) to screen out Group write permission, so that files created on behalf of a client that has a GID other than -1 are created with a permission of read (r--) instead of read and write (rw-). This would mean that only the owner of the file could write that file; other members of the owner's Group would only be able to read it.

### J.11.4 Messages

Error messages may include:

Syntax: CREATION NO or pathname

Pathname specified for CREATION is not valid

---

## J.12 DROPFILE Subcommand

▶▶ DROPFILE *pathname* ◀◀

### J.12.1 Purpose

Use the DROPFILE subcommand to remove a file from cache after the current usage is complete.

### J.12.2 Operands

*pathname*

is the absolute path name of the file that should be dropped.

### J.12.3 Usage Notes

1. If the specified file is currently in use and is not in the permanent file list, that file will be removed from cache after the current usage is complete.  
  
If the specified file is currently in use and is in the permanent file list, that file is marked as having been dropped. The first new request for access to this file will cause a new copy to be obtained from the Byte File System.
2. A file can be requested by a client in two different forms, either NETASCII or OCTET; both forms of the file can reside in cache at the same time. In such a case, only the first instance of the file encountered in the cache will be dropped by the DROPFILE subcommand.

---

## J.13 EXIT Subcommand

```
▶▶—EXIT—▶▶
```

### J.13.1 Purpose

Use the EXIT subcommand to stop the TFTP daemon. This subcommand is equivalent to the **QUIT** and **STOP** subcommands.

---

## J.14 HELP Subcommand

```
▶▶—HELP—▶▶
```

### J.14.1 Purpose

Use the HELP subcommand to display a brief description of available subcommands.

---

## J.15 LIGHT Subcommand

```
▶▶—LIGHT—▶▶
```

### J.15.1 Purpose

Use the LIGHT subcommand to toggle the light-traffic mode in the TFTP daemon.

### J.15.2 Usage Notes

1. In light-traffic mode, the daemon works on the assumption that it will process requests faster than they will queue up. In this mode, the server will receive one datagram and then return to wait for an indication that another datagram has arrived.

In heavy-traffic mode, the daemon works on the assumption that most often, it will be kept busy by a queue of datagrams. In this mode, the server will continue attempts to receive additional datagrams until it gets a “no more datagrams” response (EWOULDBLOCK error value), and then waits for another datagram to arrive.

### J.15.3 Responses

1. The following is displayed upon completion of this subcommand:

LIGHT is now *t*

where *t* is 0 if LIGHT mode is off; 1 if LIGHT mode is on.

---

## J.16 LOADPERM Subcommand

▶▶—LOADPERM—◀◀

### J.16.1 Purpose

Use the LOADPERM subcommand to reload a copy of the permanent file list, TFTPDP PERMLIST.

### J.16.2 Usage Notes

1. Files listed in the permanent file list (TFTPDP PERMLIST) are kept in memory after an initial access by a client. There is no provision to “pre-load” a file into cache before it is requested by a client.
2. The TFTPDP PERMLIST file format is:
  - One line per pathname.
  - Blank lines and lines where the first word is a “\*” are ignored by TFTPDP.
3. Maintaining files in memory improves TFTPDP performance at the expense of increased virtual storage utilization.
4. The LOADPERM subcommand loads the first occurrence of the TFTPDP PERMLIST file present in the search order.
5. Absolute path names should be used to identify files in the TFTPDP PERMLIST file.

---

## J.17 LOADUSER Subcommand

▶▶—LOADUSER—◀◀

## J.17.1 Purpose

Use the LOADUSER subcommand to load a copy of the user mapping file, TFTPUSERLIST.

## J.17.2 Usage Notes

1. The user mapping file (TFTPUSERLIST) associates client IP addresses with a VM system user ID. This mapping allows an administrator to control user access to various files via the POSIX user ID (UID) and group ID (GID) permissions that correspond to the VM user IDs identified within this file.
2. The TFTPUSERLIST file format is:
  - One line per IP address and user ID pair. This line consists of two blank delimited words, where the:
    - first word is the client IP address, in dotted decimal notation.
    - second word is the VM user ID that should be associated with this IP address. A user ID may be associated with more than one client IP address.
  - Blank lines and lines where the first word is a "\*" are ignored by TFTP.
3. The LOADUSER subcommand loads the first occurrence of the TFTPUSERLIST file present in the search order.

---

## J.18 QUIT Subcommand

```
▶▶—QUIT—▶▶
```

### J.18.1 Purpose

Use the QUIT subcommand to stop the TFTP daemon. This subcommand is equivalent to the **EXIT** and **STOP** subcommands.

---

## J.19 STAYUP Subcommand

```
▶▶—STAYUP—▶▶
```

## J.19.1 Purpose

Use the STAYUP subcommand to toggle the STAYUP mode in the TFTP daemon. If the daemon is already operating in STAYUP mode, it will cease operating in this mode and will end processing if a subsequent TCP/IP failure occurs. If the daemon is not operating in STAYUP mode, it will begin to ensure processing will not end if a subsequent TCP/IP failure occurs.

## J.19.2 Responses

1. The following is displayed upon completion of the subcommand:

```
STAYUP is now t
```

where *s* is 0 if STAYUP mode is off; 1 if STAYUP mode is on.

---

## J.20 STOP Subcommand

```
▶▶—STOP—▶▶
```

### J.20.1 Purpose

Use the STOP subcommand to stop the TFTP daemon. This subcommand is equivalent to the **EXIT** and **QUIT** subcommands.

---

## J.21 TRACE Subcommand

```
▶▶—TRACE—▶▶
```

### J.21.1 Purpose

Use the TRACE subcommand to toggle the TRACE mode in the TFTP daemon. If the daemon is already operating in TRACE mode, it will cease displaying debug information as it processes requests. If the daemon is not operating in TRACE mode, it will display debug information as it processes requests.

### J.21.2 Responses

1. The following is displayed upon completion of this subcommand:

```
TRACE is now t
```

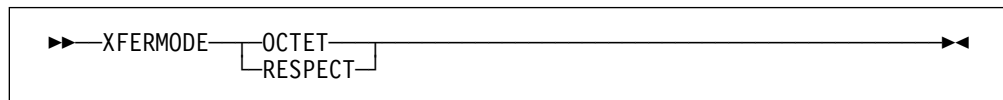
where *t* is 0 if TRACE mode is off; 1 if TRACE mode is on.

2. See Appendix N, “TFTPD Trace Records” on page 352 for a description of TFTPD server trace output.



---

## J.22 XFERMODE Subcommand



### J.22.1 Purpose

Use the XFERMODE subcommand to indicate whether TFTPD should translate data that is sent or received with a transfer mode of NETASCII.

### J.22.2 Operands

#### OCTET

indicates that TFTPD should ignore the transfer mode requested by the client. No translation will be performed on the transferred data. This allows data to be stored in the Byte File System in ASCII format.

#### RESPECT

indicates that TFTPD should respect the transfer mode requested by the client. All data in the Byte File System is assumed to be EBCDIC data. When the client requests that the file be transferred as NETASCII, the data is converted between EBCDIC and NETASCII.

### J.22.3 Messages

Error messages may include:

Syntax: XFERMODE OCTET or RESPECT

---

## J.23 TFTP File Access Control

File access for clients served by TFTP is controlled by the Byte File System (BFS). A requester can only access files in the BFS subdirectory tree of the BFS directory mounted by the TFTP server at initialization. Additionally, the VM user ID that corresponds to a client must have the proper permission, so that the client can access these files.

For BFS files, there are three classes of users whose access can be controlled:

- Owner (the owner of a file or directory, whose UID matches the UID for the file)
- Group (a member of any group whose GID matches the GID for the file)
- Other (anyone else)

File access for these groups is controlled via a set of *permission* bits.

To perform a read or write task, a user ID have been granted the proper permission in at least one of these classes.

| Usually, TFTP clients requesting access to Byte File System files are handled as  
| "Other," or public, requestors. However, it is possible to associate clients with a  
| VM user ID so that additional access protection can be provided by the TFTP  
| daemon. Once a client is associated with a specific user ID, that user ID's Owner  
| (UID) and Group (GID) access is used to determine whether that client may access  
| a file.

| The TFTP USERLIST file is used to associate, or map, client IP addresses to a  
| VM user ID. This file is loaded when the TFTP server is initialized, and can be  
| reloaded by the **LOADUSER** subcommand. See J.5, "Create the TFTP  
| USERLIST Data File" on page 317 and J.17, "LOADUSER Subcommand" on  
| page 328 for more information about the TFTP USERLIST file and how to reload  
| it.

---

## Appendix K. Processing the IBM Network Station Client Code

**Note!**

This appendix consolidates information previously found in the NSTATION README file provided with APAR PQ01770, and the TCP/IP V2 R4 for VM Corrective Service Memo.

Additionally, service information has been added for the following APARs:

- PQ05644
- PQ06451
- PQ08053
- PQ08331
- PQ12439

In order to function, the IBM Network Station loads various files from the Byte File System (BFS) into memory, via TFTP. These files (the "Client Code") are provided via APAR PQ12439 as a system "tar" file, NSTATION TARBIN. This appendix describes how to process this file and configure the IBM Network Station Client Code for your environment.

The NSTATION TARBIN file can be found on the P735FALK 493 minidisk after you have completed installing the service provided via APAR PQ12439.

**Note:** The NSTATION TARBIN file is a large file (approximately 10.6 Megabytes (M); approximately 21M when in CMS "packed" format). Thus, the time required for VMSES/E to process this file will be somewhat longer than for typical TCP/IP V2 R4 for VM files.

---

### K.1 IBM Network Station Service Updates

This section provides summary information about changes incorporated in the "Client Code" through the service process.

#### K.1.1 Service Level Information

The most current service level of the NSTATION TARBIN file is available via APAR PQ08831. The changes incorporated within this, and any previous APARs, are described in the following section.

## K.1.2 Service Level Information For APAR PQ05644

NSTATION TARBIN file changes provided through APAR PQ05644 are summarized in Figure 56.

Figure 56 (Page 1 of 3). NSTATION TARBIN Fixes - APAR PQ05644

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ03753	SA63176	CHANGES FOR NS3270 (REFLECT MULTIPLE SESSIONS)
PQ03754	SA63193	3270 COPY/PASTE PANIC
PQ03755	SA63280	3270 "KEYFILE: %4" WHEN USING DEFAULT.101 KEYBD FILE
PQ03756	SA63281	3270 PANIC DURING CUT/PASTE IN MULTI-SESSIONS
PQ03757	SA63282	MRI_PATH NOT BEING USED BY 3270
PQ03758	SA63283	MRI_PATH INTERFACE CHANGES FOR HELP/KEYMAPPER
PQ03759	SA63284	3270 HELP NOT DISPLAYED
PQ03760	SA63285	3270 PROVIDE SELECTION OF DIFFERENT MOD TYPES FOR NEW SESSION
PQ03761	SA63286	3270 SOME COMMAND LINE OPTIONS NOT WORKING
PQ03762	SA63287	3270 TRACE NOT WORKING
PQ03763	SA63288	3270 DON'T ALLOW GRAPHICS CASCADING MENU TO HAVE MOD OPTIONS
PQ03764	SA63289	3270 COPY/PASTE PANIC
PQ05275	SA64689	3270 PANIC WITH 2 HOSTNAME PROMPT BOXES UP
PQ05277	SA64690	3270 TRACE WITH GRAPHICS RUNNING HANGS SESSION
PQ05279	SA64691	3270 MULTI-SESSION AUTOSTART PROB.
PQ05281	SA64692	ENABLE 3270 TO ACCESS ADDITIONAL LANGID FILES
PQ05490	SA65245	3270 HELP TEXT: CLARIFY LANGUAGE INFORMATION
PQ03750	SA62786	5250 SYSTEM DEFAULT COLOR SCHEMES TO MAKE AVAILABLE
PQ03751	SA63229	5250 SHOULD FIND HELP FILE IN SYSTEM MRI DIRECTORY
PQ03752	SA63230	5250 SYSTEM DEFAULT COLOR SCHEMES TO MAKE AVAILABLE
PQ03765	SA63520	5250 HELP TEXT UPDATE
<p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. See K.1.3.1, "Processing Notes for PQ05127" on page 336 for detailed information about processing this APAR.</li> <li>2. See K.1.3.2, "Processing Notes for PQ05644" on page 337 for detailed information about processing this APAR.</li> </ol>		

Figure 56 (Page 2 of 3). NSTATION TARBIN Fixes - APAR PQ05644

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ05293	SA64808	5250 NEW FUNCTION AND FIXES
PQ05453	SA65203	5250 LOCAL PRINT SUPPORT ADDED
PQ05283	SA64810	5250 HELP TEXT CHANGES
PQ05127	SA64901	BOOT MONITOR VERSION 2.8.AD (1*)
PQ05644	SA65388	BOOT MONITOR VERSION 2.8.AE (2*)
PQ03877	SA62784	PANIC 8 PROPERTIES DEFINED IN JAVA APP
PQ03884	SA62949	SUPPORT FOR JAVA 8 PROPERTIES FAILURE
PQ04952	SA64415	JAVA AWT SUPPORT FOR NAVIO BROWSER
PQ05271	SA62826	JAVA APPLLET CAUSES TASKBAR PROBLEM
PQ05273	SA64919	JAVA APPLLET PARAMETERS NOT USED
PQ03874	SA62783	IBM NETWORK STATION KERNEL PANIC WITH BROWSER
PQ03880	SA62873	WINDOW MANAGER FIXES (WM.NWS)
PQ03882	SA62874	RENAME FAILS IF FILE EXIST
PQ04894	SA63890	PROBLEMS WITH NON US ENGLISH KEYBOARD FILES
PQ04896	SA63891	ENVIRONMENT VARIABLES ACCESSIBLE
PQ04816	SA64359	NEW FONT NEEDED FOR 5250 EMULATOR
PQ04818	SA64379	ALLOW LOADB TO WORK WITH NETWORKSTATION BROW
PQ04550	SA64410	TELNET HAS ERRORS ON NETWORK STATION
PQ04898	SA64624	ODD FILENAMES IN QDLS WHEN USING RFS/400
PQ05112	SA65007	ADD SUPPORT FOR SERIAL PRINTING TO PRINT APIS
PQ05500	SA65263	PROBLEMS WITH NON US ENGLISH KEYBOARD FILES
PQ03742	SA62727	LOGIN DESKTOP PREFERENCES ARE NOT USED
PQ03743	SA62831	LOGIN USER PROFILE IN LOWER CASE W/O PREFERENCES SET
PQ03744	SA62833	PROBLEMS AT LOGIN SCREEN
PQ03745	SA62834	LOGIN TOP/BOTTOM OF SCREEN TOGGLE
PQ03746	SA62835	JAVA CLOCK HAS WRONG DATE/TIME
<p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. See K.1.3.1, "Processing Notes for PQ05127" on page 336 for detailed information about processing this APAR.</li> <li>2. See K.1.3.2, "Processing Notes for PQ05644" on page 337 for detailed information about processing this APAR.</li> </ol>		

Figure 56 (Page 3 of 3). NSTATION TARBIN Fixes - APAR PQ05644

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ03747	SA62852	LOGIN TOP OF SCREEN/HIDE VALUES-SAVE AFTER LOG OFF
PQ03748	SA62853	LOGIN - LOGOUT - PANIC
PQ03824	SA62945	LOGIN MESSAGE DATA NOT FOUND
PQ03749	SA63077	LOGIN - LOGOUT - PANIC
PQ05257	SA64506	ACTLOGIN SHOULD HONOR GEMOMETRY PARAMETERS
PQ05261	SA64508	LOGIN ENTER KEY AT RIGHT HAND SIDE NOT WORKING.
PQ05263	SA64509	LOGIN READS 5250 PREF + PANICS IF > 7 COLOR SCHEMES
PQ05265	SA64510	LOGIN CLIENT ALLOWS ENTRY OF USERIDS OF LENGTH >10 CHARS
PQ05267	SA64511	LOGIN SCREEN STAYED LONG WHEN SIGN ON NS.
PQ05269	SA64512	LOGIN CLIENT MUST SUPPORT USERNAME CASE NORMALIZATION
PQ03766	SA64813, SA63524	LOGIN MESSAGE IS INCORRECT.
PQ05827	SA65545	LOGIN PREF FILES READ FROM WRONG HOST
<b>Notes:</b> <ol style="list-style-type: none"> <li>1. See K.1.3.1, "Processing Notes for PQ05127" for detailed information about processing this APAR.</li> <li>2. See K.1.3.2, "Processing Notes for PQ05644" on page 337 for detailed information about processing this APAR.</li> </ol>		

## K.1.3 Additional Processing Notes - APAR PQ05644

### K.1.3.1 Processing Notes for PQ05127

After this PTF is applied, you should use the IBM Network Station Manager to setup the automatic load of the 2.8.AD Boot Monitor image.

1. Using a browser or your IBM Network Station browser, access the IBM Network Station Manager for VM.
2. Select "Hardware" under Setup Tasks.
3. Select the proper Hardware Defaults. (System, WorkStation or User).  
Selecting System defaults is recommended. Boot Monitor 2.8.AD is supported on the following IBM Network Station models: 100, 200, 110, and 210.
4. Use the "Update boot monitor from this file:" input field to select file "bp28ad."
5. Select "Finish" to apply the configuration.

6. Each IBM Network Station using VM/ESA as its server must be restarted (That is, power off, and then power on each unit).
7. After the IBM Network Stations have been updated, use the IBM Network Station Manager to set the "Update boot monitor from this file:" selection to "Default (no update)."

**Note:** After installing this PTF, file "bp28ac" will not exist in the /QIBM/ProdData/NetworkStation/proms directory. Boot monitor image "bp28ad" should be used on all IBM Network Stations. "bp28ac" will not be supported in the near future, so all IBM Network Stations should use "bp28ad."

### K.1.3.2 Processing Notes for PQ05644

After this PTF is applied, you should use the IBM Network Station Manager to setup the automatic load of the 2.8.AE Boot Monitor image.

1. Using a browser or your IBM Network Station browser, access the IBM Network Station Manager for VM.
2. Select "Hardware" under Setup Tasks.
3. Select the proper Hardware Defaults. (System, WorkStation or User). Selecting System defaults is recommended. Boot Monitor 2.8.AE is supported on the following IBM Network Station models: 100, 200, 110, and 210.
4. Use the "Update boot monitor from this file:" input field to select file "bp28ae."
5. Select "Finish" to apply the configuration.
6. Each IBM Network Station using VM/ESA as its server must be restarted (That is, power off, and then power on each unit).
7. After the IBM Network Stations have been updated, use the IBM Network Station Manager to set the "Update boot monitor from this file:" selection to "Default (no update)."

### K.1.4 Service Level Information For APAR PQ06451

NSTATION TARBIN file changes provided through APAR PQ06451 are summarized in Figure 57.

Figure 57 (Page 1 of 2). NSTATION TARBIN Fixes - APAR PQ06451

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ06396	SA65993	3270 PRODUCT FIXES
PQ06628	SA66141	UPDATE HELP TEXT
PQ06451	SA66035	NEW KEYBOARD REMAPPING PROGRAM
PQ06630	SA66142	5250 HELP TEXT
PQ06457	SA66045	NSTATION PROBLEMS WHEN LOGON CANCELLED

Figure 57 (Page 2 of 2). NSTATION TARBIN Fixes - APAR PQ06451

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ06459	SA66046	LOGOUT/LOGIN DOES NOT WORK WITH BOOTP STARTUP
PQ06461	SA66047	NO HELP WITH COMMON (RAP) LOGIN CLIENT

### K.1.5 Service Level Information For APAR PQ08053

NSTATION TARBIN file changes provided through APAR PQ08053 are summarized in Figure 58.

Figure 58. NSTATION TARBIN Fixes - APAR PQ08053

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ06969	SA66364	MESSAGES DURING NS LOGIN NOT BEING DISPLAYED.
PQ06976	SA66374	HELP FILES NEEDED FOR NT AND MVS, MESSAGE FIXED TOO.
PQ07261	SA66607	ADDITION OF USER GROUP DEFAULT
PQ07452	SA66758	MRI RESOURCE UPDATE FOR ACTLOGIN
PQ07390	SA66699	JAVA DOES NOT SUPPORT HTTP 1.1
PQ07398	SA66711	JAVA FONTS AND TIMEZONES INCORRECT
PQ07508	SA66801	UNPRED SCREEN CORRUPTION
PQ07651	SA66925	JAVA FONTS AND TIMEZONES INCORRECT
PQ08008	SA67238	NETWORK STATION 5250 MAXIMIZE ENHANCE
PQ08021	SA66614	5250 FOCUS ERRANT BEHAVIOR. NEW WINDOW OPENS WITH FOCUS FROM COMMAND-NEW 5250 BUT CANNOT ENTER DATA.
PQ08132	SA67275	5250 PLYBCK DOESN'T WORK IN NUMLK
PQ08053	SA67274	NSCONSOLE/400 ENHANCEMENT
PQ08057	SA67277	LOCK SCREEN BUTTON MISSING
PQ08073	SA67280	ERRATIC TASKBAR SCROLLBAR ACTION
PQ08075	SA67284	NCDTERM RESOURCE FILE ACCESS
PQ08372	SA67552	3270 PRODUCT FIXES
PQ08397	SA67567	UPDATE HELP TEXT FOR 3270



### K.1.6 Service Level Information For APAR PQ08331

NSTATION TARBIN file changes provided through APAR PQ08331 are summarized in Figure 59.

Figure 59. NSTATION TARBIN Fixes - APAR PQ08331

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ08331	SA67527	NEW FUNCTION FOR RELEASE 2.5
PQ08380	SA67558	NEW FUNCTION FOR RELEASE 2.5
PQ08411	SA67580	NEW FUNCTION FOR RELEASE 2.5
PQ08416	SA67582	NEW FUNCTION FOR RELEASE 2.5
PQ08439	SA67595	NEW FUNCTION FOR RELEASE 2.5
PQ08467	SA67607	NEW FUNCTION FOR RELEASE 2.5
PQ08526	SA67164	NEW FUNCTION FOR RELEASE 2.5
PQ09508	SA67164	NEW FUNCTION FOR RELEASE 2.5
PQ09652	SA68695	PRINT API PTF FOR REL 2.5

### K.1.7 Service Level Information For APAR PQ12439

NSTATION TARBIN file changes provided through APAR PQ12439 are summarized in Figure 60.

Figure 60 (Page 1 of 2). NSTATION TARBIN Fixes - APAR PQ12439

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ10685	SA69569	NETWORK STATION BOOT MONITOR DECOMPRESSION SUPPORT
PQ09747	SA68794	NTWRKSTN-UNPRED EXTENSIONS FOR MAINTAINABILITY
PQ10663	SA69555	NEW FUNCTION
PQ10667	SA68164	NEW FUNCTION
PQ10669	SA69556	NEW FUNCTION
PQ10671	SA69557	NEW FUNCTION
PQ10673	SA69558	NEW FUNCTION
PQ10675	SA69561	NEW FUNCTION
PQ10677	SA69562	NETWORK STATION INTERNAL MAINTENANCE
PQ10679	SA69564	NETWORK STATION INTERNAL MAINTENANCE
PQ10681	SA69559	NEW FUNCTION

Figure 60 (Page 2 of 2). NSTATION TARBIN Fixes - APAR PQ12439

TCP/IP for VM APAR Number	Sysroute of Network Station APAR	Description
PQ10683	SA69565	NEW FUNCTION
PQ10690	SA69575	NEW FUNCTION
PQ10692	SA69576	NEW FUNCTION
PQ10696	SA69577	NEW FUNCTION
PQ10698	SA69578	NEW FUNCTION
PQ10700	SA69579	NEW FUNCTION
PQ10703	SA69580	NEW FUNCTION
PQ11040	SA69868	JAVA PRINT API
PQ11416	SA70218	NTWRKSTN-UNPRED EXTENSIONS FOR MAINTAINABILITY
PQ11424	SA70225	NTWRKSTN-UNPRED EXTENSIONS FOR MAINTAINABILITY
PQ11427	SA70226	NTWRKSTN-UNPRED EXTENSIONS FOR MAINTAINABILITY
PQ11430	SA70231	NTWRKSTN-UNPRED EXTENSIONS FOR MAINTAINABILITY
PQ11434	SA70232	NTWRKSTN-UNPRED EXTENSIONS FOR MAINTAINABILITY
PQ11436	SA70234	NTWRKSTN-UNPRED EXTENSIONS FOR MAINTAINABILITY
PQ11696	SA70410	NEW FUNCTION
PQ11698	SA70411	NEW FUNCTION
PQ12006	SA70460	NTWRKSTN-INCORROUT PRODUCT DID NOT SHIP WITH COMPLETE SET OF FONTS
PQ12262	SA70912	NEW FUNCTION
PQ12439	SA71045	NETWORK STATION UP COMMAND PROBLEM

## K.2 IBM Network Station Processing Requirements

Before you complete the steps described in K.4, "Process the NSTATION TARBIN File" on page 342, you must first ensure the following conditions have been met:

- Sufficient Byte File System file space is available to accept the "Client Code" files when the NSTATION TARBIN file is "exploded," or processed. The space required for the resulting files is approximately **8000** 4K blocks.
- For the P735FALK user ID, ensure the following:
  - sufficient virtual storage available to process the NSTATION TARBIN "tar" file; at least **64M** of virtual machine storage is required.

- this user ID is defined as a POSIX “superuser.” To allow this capability, the following statement must be included in the P735FALK user ID directory entry:

```
POSIXINFO UID 0 GID 0
```

See the *OpenEdition for VM/ESA User's Guide* and *VM/ESA Planning and Administration* for more information about defining user IDs in this manner.

- this user ID is defined with administrator authority for the BFS file pool where the “Client Code” will be placed. See *VM/ESA File Pool Planning, Administration and Operation* for details about SFS/BFS file pool configuration and administrator authority.
- the CMS **LOADBFS** exec is present in the CMS search order. By default, this exec resides on the MAINT 193 minidisk. You may need to link and access this (or some other) minidisk so that the LOADBFS exec can be used.
- The OpenEdition Shell and Utilities Feature for VM/ESA (the OpenEdition for VM/ESA “shell”) is installed and is available to the P735FALK user ID.

**Note!**

If you use an external security manager (ESM) in your environment, see your ESM product documentation for additional information that may apply to user ID (UID) and group ID (GID) usage.

### K.3 NSTATION LOADBFS Modifications for Installing in a Non-default File Pool

If you plan to install IBM Network Station “Client Code” files in a Byte File System file pool different from VMSYSU (the default), you will need to update the NSTATION LOADBFS file to reflect the file pool you want to use. In so doing, you'll need to replace all “VMSYSU” references within this file with your file pool name.

Additionally, if you installed the Byte File System root file space and root directory (and thus, the OpenEdition Shell and Utilities Feature for VM/ESA) in a file pool different from the VM/ESA default (VMSYS), you will also need to update NSTATION LOADBFS references to “VMSYS,” to reflect the BFS root file space defined for your system.

For more information about installing the “Client Code” files in a non-default file pool, see “BFS Root File Space in a Non-default File Pool” in *VM/ESA File Pool Planning, Administration and Operation*.

If you need to modify the NSTATION LOADBFS file, you should do so *before* you begin the steps described in K.4, “Process the NSTATION TARBIN File” on page 342.

It's recommended that you make your changes the NSTATION \$LOADBFS file, rather than to the NSTATION LOADBFS file itself, using the VMSES/E local modification process. This will allow service-related changes to this file to be reported during the VMSES/E service process. For information about installing local modifications, see the *VMSES/E Introduction and Reference*.

**Note:** If you install the “Client Code” files in a non-default file pool, you will also need to change the default BFS directory which is mounted by the TFTP server. See Appendix J, “Configuring the TFTP Virtual Machine” on page 314 for more information about how to change this default.

---

## K.4 Process the NSTATION TARBIN File

The steps that follow describe how to set up the required file space and add the IBM Network Station data to the Byte File System.

### Important Note!

The NSTATION TARBIN file is provided in “packed” format (fixed format, with a logical record length of 1024) to allow for proper VMSES/E processing. You must unpack the NSTATION TARBIN file, as described in step 2 below, before this file can be processed using the remainder of the steps described in this section.

1. Logon the installation user ID, P735FALK.
2. **Unpack** the NSTATION TARBIN file.

**copyfile nstation tarbin *fm* = = = (unpack olddate replace**

*fm* is the file mode of the P735FALK 493 minidisk.

3. Run the LOADBFS command against the NSTATION LOADBFS file.

**loadbfs nstation**

The **LOADBFS** command will create the QIBM BFS file space in the appropriate file pool (VMSYSU, by default). It will also create a Mount External Link (MEL) called /QIBM and write the NSTATION TARBIN file out to the VMBFS:VMSYSU:QIBM BFS file space.

- |  
| 4. Mount the Byte File System root directory which contains the OpenEdition Shell  
| and Utilities Feature for VM/ESA.  
|

— **Note!** —

The commands and operands used throughout the remainder of this procedure are case sensitive; these commands must be issued with respect to the case in which they appear.

| **openvm mount** *./VMBFS:filepool:ROOT/* /

| Specify *filepool* as **VMSYS** if the OpenEdition Shell  
| and Utilities Feature for VM/ESA is installed in the  
| VM/ESA default BFS file pool. Otherwise, specify  
| the file pool name of the BFS root file space  
| defined for your system. Note that *filepool* must  
| be specified using upper case.

- |  
| 5. Start an OpenEdition for VM/ESA shell and enter the shell command  
| environment.

| **openvm shell**

- |  
| 6. If you are re-exploding the “tar” file into a structure that you previously created,  
| you must remove the previous version of this structure.

| **rm -Rf /QIBM/ProdData/**

- |  
| 7. Explode the “tar” file so that the necessary BFS directory structure is created,  
| and that all files are placed in the appropriate directories.

| **pax -rzf /nets.tar**

- |  
| 8. Remove the “tar” file from the Byte File System.

| **rm /nets.tar**

- |  
| 9. Exit the OpenEdition for VM/ESA shell.

| **exit**

---

## K.5 IBM Network Station Customization

### K.5.1 IBM Network Station Configuration Files

**Note:** The IBM Network Station Manager for VM/ESA, should be used to configure your IBM Network Station networking environment.

For information on how to obtain the IBM Network Station Manager and IBM Network Station Login Server (NSLD) code, see the VM IBM Network Station home page at the following URL:

<http://www.vm.ibm.com/NetworkStation>

Also, pay close attention to the following notes:

- Users must access the IBM Network Station Manager via Netscape to initially set options from the “setup tasks” menu as an administrator. After this has been done, the IBM Network Station Manager can be accessed by users via the IBM Network Station Browser.
- Before saving anything from within a 3270 or 5250 emulation session, or the IBM Network Station Browser, a user must first save preferences for each application using the IBM Network Station Manager. This is necessary because the directory these applications attempt to write to will not exist until preferences have been saved.
- If using the IBM Network Station Manager causes files to be locked — even after completion, refresh the Web Server before starting your IBM Network Station.

### K.5.2 IBM Network Station BOOT Configuration

Before you boot an IBM Network Station, ensure that the BOOTPD, TFTP and NSLD servers have been initialized and properly function. Specifically, the IBM Network Station requires that the TFTP server run with the **XFERMODE OCTET** command option in effect, so that it will not attempt to translate files that are requested as NETASCII. Also, start the TFTP server with the **CREATION /QIBM** command option.

To boot an IBM Network Station with this support in place, attach it to your network and power it on. If this is the first time you are booting the station, hit the **Escape** key during the boot sequence to get to the “IBM Network Station Setup Utility” screen (**SCRN02** will be displayed in the upper left corner of the display for this screen), and use the following steps to configure the IBM Network Station.

1. From **SCRN02**, hit **F5** to get to the “Set Network Parameters” screen (**SCRN04**), shown in Figure 61 on page 345. Use this screen to set the following:

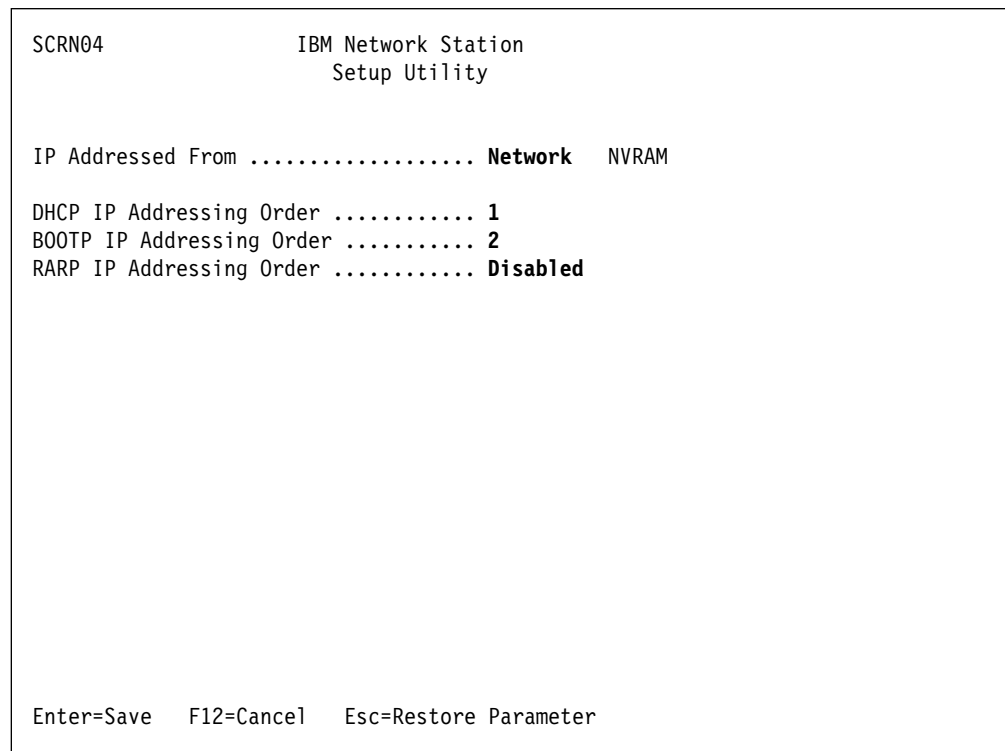


Figure 61. IBM Network Station Setup Utility Screen

When finished, hit **Enter** to save these settings, and return to the “IBM Network Station Setup Utility” (SCRN02).

2. Again, from SCRNO2, hit **F6** to get to the “Set Boot Parameters” screen (SCRNO6), shown in Figure 62 on page 346. Use this screen to set the following:

```

SCRN06                IBM Network Station
                        Set Boot Parameters

Boot File ..... (leave blank)

TFTP Boot Directory ..... /QIBM/ProdData/NetworkStation/

NFS Boot Directory ..... (leave blank)

Configuration File ..... (leave blank)

Configuration Directory ..... /QIBM/ProdData/NetworkStation/
                               StationConfig/
TFTP Order ..... 1
NFS Order ..... Disabled
MOP Order ..... Disabled
Local Order ..... Disabled

Enter=Save  F12=Cancel  Esc=Restore Parameter

```

Figure 62. IBM Network Station Set Boot Parameters Screen

**Note:** If service has been applied to the “Client Code” files, the Configuration Directory shown in Figure 62 should be specified as:

**/QIBM/ProdData/NetworkStation/configs/**

When finished, hit **Enter** to save these settings, and return to the “IBM Network Station Setup Utility” (SCRN02).

3. From SCRN02, you can now hit **Enter** to reboot the IBM Network Station with the parameters you just specified.



---

## Appendix L. BOOTPD Trace Records

BOOTPD trace entries identify 5 basic events:

- Time at which BOOTPD began processing a set of requests
- Reception of a datagram from a client or gateway
- Declining to respond to a client or gateway, due to some error or limit
- Forwarding of a request to another BootP daemon
- The attempt to respond to a client or gateway

---

### L.1 BOOTPD Trace Record Format

The first line of a trace entry consists of a trace code followed by a description of the event, along with other pertinent information. Additional lines of information may be displayed, indented under the first line.

---

### L.2 BOOTPD Trace Codes

The BOOTPD trace codes are:

<b>1000</b>	Received a request sent by a client
<b>1100</b>	Received a request that was forwarded by a BootP daemon
<b>1900</b>	Unrecognized request was received; the opcode was neither request or reply
<b>3000</b>	A BootP reply was sent to a client
<b>3100</b>	A BootP reply was sent to another BootP daemon, to be passed to a client
<b>32xx</b>	Request is being forwarded to another BootP daemon. The xx subcodes that follow indicate the reasons for forwarding:
<b>01</b>	Forwarding was specified, but no entry exists in the machine table
<b>02</b>	Always forward was specified
<b>03</b>	Client specified a server to which to forward the request
<b>00</b>	Reason for forwarding was not known
<b>40xx</b>	The BootP daemon is declining to respond to a request. The xx subcodes that follow indicate the reasons for declining to respond:
<b>01</b>	Entry was not found in the machine table
<b>02</b>	Request received on an adapter that was partially excluded, for which the entry matches the exclusion criteria
<b>03</b>	Unrecognized packet opcode was received
<b>04</b>	Could not forward because the hop count expired
<b>05</b>	Could not determine the client IP address
<b>06</b>	Could not determine the bootfile pathname
<b>07</b>	Target server is on the same cable

	<b>08</b>	Unable to determine the adapter over which to reply
	<b>00</b>	Reason for declining is not known
<b>9000</b>		Time Stamp, including the time and date in standard format

Trace events which relate to the transmission of BOOT requests or replies, include information about the packet.

OP	=	<i>opcode</i>	CIADDR	=	<i>ipaddr</i>
HTYPE	=	<i>htype</i>	YIADDR	=	<i>ipaddr</i>
HLEN	=	<i>hlen</i>	SIADDR	=	<i>ipaddr</i>
HOPS	=	<i>hops</i>	GIADDR	=	<i>ipaddr</i>
XID	=	<i>xid</i>	CHADDR	=	<i>chaddr</i>
SNAME	=	<i>servname</i>			
FILE	=	<i>bootfile</i>			
VEND	=	<i>venddata</i>			

where

**OP** = *opcode*  
indicates the operation code: 1 for a request or 2 for a reply.

**CIADDR** = *ipaddr*  
indicates the client IP address, if specified by the client.

**HTYPE** = *htype*  
indicates the network hardware type.

**YIADDR** = *ipaddr*  
indicates the IP address of the client.

**HLEN** = *hlen*  
indicates the length of the hardware address.

**SIADDR** = *ipaddr*  
indicates the Server IP address.

**HOPS** = *hops*  
indicates the current hop count.

**GIADDR** = *ipaddr*  
indicates the gateway IP address.

**XID** = *xid*  
indicates the current transaction ID specified by the client.

**CHADDR** = *chaddr*  
indicates the client hardware address. This field may be a maximum of 16 bytes long.

**SNAME** = *servname*  
indicates the Server Host Name. This field may be a maximum of 64 bytes long.

**FILE** = *bootfile*  
indicates the boot file name. This field may be a maximum of 128 bytes long.

**VEND** = *venddata*  
indicates the current contents of the vendor-specific area. This field may be a maximum of 64 bytes long.

---

## / Appendix M. DHCPD Trace Records

/ DHCPD trace entries identify 6 basic events:

- / • Time at which DHCPD began processing a set of requests
- / • Reception of a datagram from a client or gateway
- / • Declining to respond to a client or gateway, due to some error or limit
- / • Forwarding of a request to another DHCPD/BootP daemon
- / • The attempt to respond to a client or gateway
- / • Timer expiration and related activities

---

### / M.1 DHCPD Trace Record Format

/ The first line of a trace entry consists of a trace code followed by a description of  
/ the event, along with other pertinent information. Additional lines of information  
/ may be displayed, indented under the first line.

---

### / M.2 DHCPD Trace Codes

/ The DHCPD trace codes are:

/	<b>1000</b>	Received a request sent by a client
/	<b>1100</b>	Received a request that was forwarded by a BootP daemon
/	<b>1900</b>	Unrecognized request was received; the opcode was neither request or / reply
/	<b>3000</b>	A BootP/DHCP reply was sent to a client
/	<b>3100</b>	A BootP/DHCP reply was sent to another BootP/DHCP daemon, to be / passed to a client
/	<b>32xx</b>	Request is being forwarded to another BootP/DHCP daemon. The xx / subcodes that follow indicate the reasons for forwarding:
/	<b>01</b>	Forwarding was specified, but no entry exists in the machine / table
/	<b>02</b>	Always forward was specified
/	<b>03</b>	Client specified a server to which to forward the request
/	<b>00</b>	Reason for forwarding was not known
/	<b>40xx</b>	The DHCP daemon is declining to respond to a request. The xx / subcodes that follow indicate the reasons for declining to respond:
/	<b>01</b>	Entry was not found in the machine table
/	<b>02</b>	Request received on an adapter that was partially excluded, / for which the entry matches the exclusion criteria
/	<b>03</b>	Unrecognized packet opcode was received
/	<b>04</b>	Could not forward because the hop count expired
/	<b>05</b>	Could not determine the client IP address
/	<b>06</b>	Could not determine the bootfile pathname

```

/
/      07      Target server is on the same cable
/      08      Unable to determine the adapter over which to reply
/      09      SupportBootP is NO
/      10      Client is on a different subnet than the requested address
/      11      Requested address is restricted
/      12      Requested address is in use by another client
/      13      Internal error
/      14      Requested address differs from machine table entry
/      15      No address is available
/      16      SupportUnlistedClients is NO
/      17      Client is not recognized
/      18      Client is not in a valid state
/      19      Request is not correctly formatted
/      20      Not selected as the server
/      21      Ignore any DHCP Offer messages
/      22      Address is being declined
/      23      Address is being released
/      24      Ignore any DHCP Ack messages
/      25      Ignore any DHCP Nack messages
/      26      Nothing possible for DHCP Inform
/      27      Client statement specified: NONE
/      28      Waiting for ICMP Echo to complete
/      00      Reason for declining is not known
/
/      5000    ICMP Timer expired with a response reply due
/      5100    Received an ICMP Echo reply
/      5300    Sending an ICMP Echo request
/      5500    Lease expired for an address
/      9000    Time Stamp, including the time and date in standard format

```

Trace events which relate to the transmission of BOOT requests or replies, include information about the packet.

```

/
/      OP      = opcode    CIADDR = ipaddr  DHCPTYPE = msgtype
/      HTYPE   = htype    YIADDR = ipaddr
/      HLEN    = hlen     SIADDR = ipaddr
/      HOPS    = hops     GIADDR = ipaddr
/      XID     = xid      CHADDR = chaddr
/      SNAME   = servname
/      FILE    = bootfile
/      OPTIONS = optiondata

```

where

```

/
/      OP = opcode
/      indicates the operation code: 1 for a request or 2 for a reply.
/
/      CIADDR = ipaddr
/      indicates the client IP address, if specified by the client.

```

/ **DHCPTYPE** = *msgtype*  
/ indicates the type of DHCP message. This parameter is shown only for DHCP  
/ protocol requests and replies.  
/ **HTYPE** = *htype*  
/ indicates the network hardware type.  
/ **YIADDR** = *ipaddr*  
/ indicates the IP address of the client.  
/ **HLEN** = *hlen*  
/ indicates the length of the hardware address.  
/ **SIADDR** = *ipaddr*  
/ indicates the Server IP address.  
/ **HOPS** = *hops*  
/ indicates the current hop count.  
/ **GIADDR** = *ipaddr*  
/ indicates the gateway IP address.  
/ **XID** = *xid*  
/ indicates the current transaction ID specified by the client.  
/ **CHADDR** = *chaddr*  
/ indicates the client hardware address. This field may be a maximum of 16  
/ bytes long.  
/ **SNAME** = *servname*  
/ indicates the Server Host Name. This field may be a maximum of 64 bytes  
/ long. When "SNAME" is followed by "(O)," the field contains configuration  
/ options instead of only SNAME data. The data shown is a hexadecimal  
/ representation of the contents of the field.  
/ **FILE** = *bootfile*  
/ indicates the boot file name. This field may be a maximum of 128 bytes long.  
/ When "FILE" is followed by "(O)," the field contains configuration options  
/ instead of only FILE data. The data shown is a hexadecimal representation of  
/ the contents of the field.  
/ **OPTIONS** = *optiondata*  
/ indicates the current contents of the vendor-specific area. This field may be a  
/ maximum of 64 bytes long.

---

## Appendix N. TFTP Trace Records

TFTP trace entries identify 5 basic events and TCP/IP errors:

- Acceptance of a read or write request
- Resending of packets due to a timeout
- Dropping of a client due to exceeding the packet resend limit
- Sending or reception of error packets
- Socket related errors

---

### N.1 TFTP Trace Record Format

The first line of the trace entry contains:

- a 4 digit trace code,
- a description of the trace code,
- a time and date stamp, and
- client identification information (when the entry relates to a client). This information can include:
  - the IP address of the client
  - the port number used by the client
  - the user ID associated with the client

Depending upon the trace entry, additional lines of information may be displayed; such lines are indented under the first line.

The following example shows the format of the first line of a client related trace entry.

```
code xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss description of trace code
```

where:

*code*

is a 4 digit trace code.

*xxx.xxx.xxx.xxx*

is the IP address of the client in dotted decimal notation.

*port*

is the port that the client is using.

*userid*

is the user ID associated with the IP address; this association is determined by the TFTP USERLIST file. If the client IP address is not listed in this file, then “.....” is displayed.

*mm/dd/yy*

is the date portion of the timestamp, where “mm” is the month, “dd” is the day, and “yy” is the year.

*hh:mm:ss*

is the time portion of the timestamp, where “hh” is the hour (in 24 hour format), “mm” is the minutes, and “ss” is the seconds.

*description of trace code*

is a 25 character description of the trace code

---

## N.2 TFTP Codes

The TFTP trace codes are:

<b>1000</b>	A read request was accepted.
<b>1500</b>	A read operation has completed.
<b>2000</b>	A write request was accepted.
<b>2500</b>	A write operation has completed.
<b>3000</b>	Timeout; a response was resent.
<b>3500</b>	Timeout; the timeout limit was reached, and the client dropped.
<b>4000</b>	a File Not Valid response was sent.
<b>4100</b>	a Missing BLKSIZE response was sent.
<b>4200</b>	An Access Violation response was sent.
<b>4300</b>	A Bad XFER (Transfer) Mode response was sent.
<b>5000</b>	A Spurious ACK was received and has been ignored.
<b>5100</b>	An Error Datagram was received.
<b>5200</b>	An Unknown Datagram was received.
<b>6100</b>	An unexpected RECVFROM error occurred.
<b>6200</b>	An unexpected SENDTO error occurred.
<b>6300</b>	An unexpected SOCKINIT error occurred.
<b>6301</b>	An unexpected SOCKET error occurred.
<b>6302</b>	An unexpected IOCTL error occurred.
<b>6303</b>	An unexpected BIND error occurred.

### N.2.1 TFTP Trace Entry: 1000

This trace code is the result of accepting a READ request.

```
1000 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss READ REQUEST ACCEPT SENT
      x c blksize      pathname
```

The first line of this entry is explained in N.1, “TFTP Trace Record Format” on page 352. The additional line consists of:

*x* indicates the transfer mode, “N” for NETASCII and “O” for OCTET mode.

*c* is a hit or miss indicator, indicating whether the file was in cache when requested (a hit) or whether it had to be loaded (a miss).

“H” indicates that the file was in cache.

“M” indicates that the file was not in cache.

**Note:** A miss would be indicated for a file in cache that is marked for a drop by the **DROFFILE** subcommand. Subsequent read requests would require a new copy of the file to be obtained.

*blksize*

is the block size being used for the transfer.

*pathname*

is the name of the file being transferred.

## N.2.2 TFTPDP Trace Entry: 1500

This trace code is the result of receiving an ACK associated with a client read operation. The ACK indicates the client received the last packet of a transmitted file.

```
1500 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss READ COMPLETED
      PKTS=pkts      FILE SIZE=filesize
```

The first line of this entry is explained in N.1, "TFTPDP Trace Record Format" on page 352. The additional line consists of:

*pkts*

number of packets sent.

*filesize*

size of the file, in bytes.

## N.2.3 TFTPDP Trace Entry: 2000

This trace code is the result of accepting a WRITE request.

```
2000 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss WRITE ACCEPTED DATA SENT
      x      blksize      pathname
```

The first line of this entry is explained in N.1, "TFTPDP Trace Record Format" on page 352. The additional line consists of:

*x* indicates the transfer mode, "N" for NETASCII and "O" for OCTET mode.

*blksize*

is the block size being used for the transfer.

*pathname*

is the name of the file that is being transferred.



## N.2.4 TFTPDP Trace Entry: 2500

This trace code is the result of receiving the DATA packet associated with a client write request.

```
2500 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss WRITE COMPLETED
      PKTS=pkts      FILE SIZE=filesize
```

The first line of this entry is explained in N.1, "TFTPDP Trace Record Format" on page 352. The additional line consists of:

*pkts*  
number of packets sent.  
*filesize*  
size of the file, in bytes.

## N.2.5 TFTPDP Trace Entry: 3000

This trace code is the result of determining that time has expired for a client to send or receive a packet; thus, the response was resent.

```
3000 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss TIMEOUT - RESPONSE RESENT
```

The format of this entry is explained in more detail in N.1, "TFTPDP Trace Record Format" on page 352.

## N.2.6 TFTPDP Trace Entry: 3500

This trace code is the result of determining that a timeout occurred, and that the maximum number of resends was reached; thus, the client was dropped.

```
3500 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss TIMEOUT - CLIENT DROPPED
```

The format of this entry is explained in more detail in N.1, "TFTPDP Trace Record Format" on page 352.

## N.2.7 TFTPDP Trace Entry: 4000

This trace code is the result of determining that the file to be sent to the client was not valid.

```
4000 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss FILE NOT VALID RESPONSE
      pathname
```

The first line of this entry is explained in N.1, "TFTPDP Trace Record Format" on page 352. The additional line consists of:

*pathname*  
is the name of the file that was not valid.

### N.2.8 TFTPDP Trace Entry: 4100

This trace code is the result of receiving a request that contained a BLKSIZE parameter for which no value was specified.

```
4100 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss MISSING BLKSIZE RESPONSE
```

The format of this entry is explained in more detail in N.1, "TFTPDP Trace Record Format" on page 352.

### N.2.9 TFTPDP Trace Entry: 4200

This trace code is the result of receiving a read request for a file that the client was not permitted to access.

```
4200 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss ACCESS VIOLATION RESPONSE
```

The format of this entry is explained in more detail in N.1, "TFTPDP Trace Record Format" on page 352.

### N.2.10 TFTPDP Trace Entry: 4300

This trace code is the result of receiving a READ or WRITE request for which a transfer mode parameter was specified, but was not valid.

```
4300 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss BAD XFER MODE RESPONSE
```

The format of this entry is explained in more detail in N.1, "TFTPDP Trace Record Format" on page 352.

### N.2.11 TFTPDP Trace Entry: 5000

This trace code is the result of receiving an unexpected ACK; which has been ignored.

```
5000 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss SPURIOUS ACK IGNORED
```

The format of this entry is explained in more detail in N.1, "TFTPDP Trace Record Format" on page 352.

### N.2.12 TFTPDP Trace Entry: 5100

This trace code is the result of receiving an error datagram from a client.

```
5100 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss ERROR DATAGRAM RECEIVED  
      ERROR=errnum errdesc
```

The first line of this entry is explained in N.1, "TFTPDP Trace Record Format" on page 352. The additional line consists of:

*errnum*

is the error number received from the client.

| *errdesc*

| is the error description sent by the client in the error datagram.

### | **N.2.13 TFTPDP Trace Entry: 5200**

| This trace code is the result of receiving an unknown datagram.

| 5200 *xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss* UNKNOWN DATAGRAM RECEIVED

| The format of this entry is explained in more detail in N.1, “TFTPDP Trace Record Format” on page 352.

### | **N.2.14 TFTPDP Trace Entry: 6100**

| This trace code is the result of encountering an unexpected error from a SOCKET RECVFROM operation.

| 6100 *xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss* BAD RECVFROM ERROR  
| RC=*rc* ERRNO=*errno*

| The first line of this entry is explained in N.1, “TFTPDP Trace Record Format” on page 352. The additional line consists of:

| *rc* is the return code set by the RECVFROM function.

| *errno*

| is the error number set by the RECVFROM function.

### | **N.2.15 TFTPDP Trace Entry: 6200**

| This trace code is the result of encountering an unexpected error from a SOCKET SENDTO operation.

| 6200 *xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss* BAD SENDTO ERROR  
| RC=*rc* ERRNO=*errno*

| The first line of this entry is explained in N.1, “TFTPDP Trace Record Format” on page 352. The additional line consists of:

| *rc* is the return code set by the SENDTO function.

| *errno*

| is the error number set by the SENDTO function.

### | **N.2.16 TFTPDP Trace Entry: 6300**

| This trace code is the result of encountering an unexpected error from a SOCKET initialization operation.

| 6300 *xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss* BAD SOCKINIT ERROR  
| RC=*rc* REASON=*reason* SOCKETS=*socket*

The first line of this entry is explained in N.1, "TFTP Trace Record Format" on page 352. The additional line consists of:

*rc* is the return code set by the Socket Initialize function.

*reason*

is the reason code set by the Socket Initialize function.

*socket*

is the socket number (if any) returned by the Socket Initialize function.

### N.2.17 TFTP Trace Entry: 6301

This trace code is the result of encountering an unexpected error from a SOCKET SOCKET operation.

```
6301 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss BAD SOCKET ERROR
      SOCKET=socket      ERRNO=errno
```

The first line of this entry is explained in N.1, "TFTP Trace Record Format" on page 352. The additional line consists of:

*socket*

is the socket number.

*errno*

is the error number set by the SOCKET function.

### N.2.18 TFTP Trace Entry: 6302

This trace code is the result of encountering an unexpected error from a SOCKET IOCTL operation.

```
6302 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss BAD IOCTL ERROR
      RC=rc      ERRNO=errno
```

The first line of this entry is explained in N.1, "TFTP Trace Record Format" on page 352. The additional line consists of:

*rc* is the return code set by the IOCTL function.

*errno*

is the error number set by the IOCTL function.

### N.2.19 TFTP Trace Entry: 6303

This trace code is the result of encountering an unexpected error from a SOCKET BIND operation.

```
6303 xxx.xxx.xxx.xxx port (userid ) mm/dd/yy hh:mm:ss BAD BIND ERROR
      RC=rc      ERRNO=errno
```

The first line of this entry is explained in N.1, "TFTP Trace Record Format" on page 352. The additional line consists of:

|                    *rc* is the return code set by the BIND function.  
|                    *errno*  
|                    is the error number set by the BIND function.

---

## Appendix O. TFTPDATA Monitor Records

### PSPI

The TFTPDATA server contributes to the CP monitor data by using the APPLDATA call class. However, to allow this, the APPLMON directory option must be specified in the TFTPDATA server directory entry.

To begin data collection, use the CP MONITOR command to enable the APPLDATA domain, and to start event monitoring. See *VM/ESA CP Command and Utility Reference* and *VM/ESA Performance* for more information about the CP MONITOR command and performance monitoring.

The data records for servers are domain X'A' APPLDATA records, the general format of which is described in the *VM/ESA Monitor Records File*. Note that this file is located on your system's CP object disk (194) in a file named MONITOR LIST1403. Each data record is preceded by a record header, which is described in Figure 63.

Every data record consists of these parts:

- CP header data
- TFTPDATA application data

The CP header data consists of the following:

Data Item	Number of Bytes
Byte offset to application data relative to start of this record	2
Length in bytes of application data	2
User ID of the server machine (in EBCDIC)	8
Product identification (in EBCDIC). For TFTPDATA records, this field contains "5735FALTFT010100".	16
Status	1
Reserved	3

Following the CP header data is the counter data.

Figure 64 shows record layout for the TFTPDATA-supplied application data. The offset values listed are the offsets into the application data area of the monitor record (field APLSDT\_ADATA). Always use the byte offset and length fields in the standard domain 10 records to locate the start and end of the application data within the record.

Figure 64. TFTPD APPLDATA Data

Offset						
Dec	Hex	Type	Len	Name	Description	
0	(0)	SIGNED	4	mon_rrq_count	The total number of read requests, or RRQs (downloads).	
4	(4)	CHARACTER	8	mon_rrq_time	The total elapsed time spent processing read requests. Accumulated in TOD clock units.	
12	(C)	UNSIGNED	8	mon_rrq_bytes	The total number of bytes sent by the server.	
20	(14)	SIGNED	4	mon_rrq_pkts_in	The total number of UDP packets received in association with read requests.	
24	(18)	SIGNED	4	mon_rrq_pkts_out	The total number of UDP packets sent in association with read requests.	
28	(1C)	SIGNED	4	mon_rrq_misses	The total number of times a read request could not be satisfied from cache.	
32	(20)	CHARACTER	32	*	Reserved and available for IBM use.	
64	(40)	SIGNED	4	mon_wrq_counts	The total number of write requests, or WRQs (uploads).	
68	(44)	CHARACTER	8	mon_wrq_time	The total elapsed time spent processing write requests. Accumulated in TOD clock units.	
76	(4C)	UNSIGNED	8	mon_wrq_bytes	The total number of bytes received by the server.	
84	(54)	SIGNED	4	mon_wrq_pkts_in	The total number of UDP packets received in association with write requests.	
88	(58)	SIGNED	4	mon_wrq_pkts_out	The total number of UDP packets sent in association with write requests.	
92	(5C)	SIGNED	4	mon_wrq_misses	The total number of times a write request could not be satisfied from cache. This will always be zero.	
96	(60)	CHARACTER	32	*	Reserved and available for IBM use.	
128	(80)	SIGNED	4	mon_abn_count	The total number of transactions that did not complete or that abnormally ended.	
132	(84)	UNSIGNED	8	mon_abn_bytes	The total number of bytes sent with transactions that failed.	
140	(8C)	UNSIGNED	4	mon_abn_pkts_in	The total number of UDP packets received in association with failed transactions.	
144	(90)	SIGNED	4	mon_abn_pkts_out	The total number of UDP packets sent in association with failed transactions.	
148	(94)	CHARACTER	44	*	Reserved and available for IBM use.	
192	(C0)	SIGNED	4	mon_timeouts	The total number of times the TFTP server timed out while waiting for an acknowledgement.	
196	(C4)	CHARACTER	8	*	Reserved for future IBM use.	


**PSPI end**

# Reader's Comments

## TCP/IP Version 2 Release 4 for VM

You may use this form to comment about this document, its organization, or subject matter. Please understand that your feedback is of importance to IBM, but IBM makes no promises to always provide a response to your feedback.

For each of the topics below please indicate your satisfaction level by circling your choice from the rating scale. If a statement does not apply, please circle N.

RATING SCALE						
<b>very satisfied</b>					<b>very dissatisfied</b>	<b>not applicable</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>N</b>	

	Satisfaction					
Ease of product installation	1	2	3	4	5	N
Time required to install the product	1	2	3	4	5	N
Contents of program directory	1	2	3	4	5	N
Readability and organization of program directory tasks	1	2	3	4	5	N
Necessity of all installation tasks	1	2	3	4	5	N
Accuracy of the definition of the installation tasks	1	2	3	4	5	N
Technical level of the installation tasks	1	2	3	4	5	N
Installation verification procedure	1	2	3	4	5	N
Ease of customizing the product	1	2	3	4	5	N
Ease of migrating the product from a previous release	1	2	3	4	5	N
Ease of putting the system into production after installation	1	2	3	4	5	N
Ease of installing service	1	2	3	4	5	N

- Did you order this product as an independent product or as part of a package?

- Independent
- Package

What type of package was ordered?

- CustomPac
  - FunctionPac
  - SystemPac
- System Delivery Offering (SDO)
- Other - Please specify type: \_\_\_\_\_



- Is this the first time your organization has installed this product?
  - Yes
  - No
- Were the people who did the installation experienced with the installation of VM products using VMSES/E?
  - Yes
    - How many years of experience do they have? \_\_\_\_\_
  - No
- How long did it take to install this product? \_\_\_\_\_
- If you have any comments to make about your ratings above, or any other aspect of the product installation, please list them below:

---

---

---

---

---

---

---

---

---

---

Please provide the following contact information:

---

Name and Job Title

---

Organization

---

Address

---

Telephone

**Thank you for your participation.**

Please send the completed form to the following address, or give to your IBM representative who will forward it to the TCP/IP for VM Development group:

IBM Corporation  
 TCP/IP for VM Development  
 Dept. G79G  
 1701 North Street  
 Endicott, NY 13760



Program Number: 5735-FAL 5801  
5802  
5750

Printed in U.S.A.

XXXX-YYYY-ZZ

